AIAA FLIGHT SIMULATION TECHNOLOGIES CONFERENCE SEPTEMBER 7-9, 1988 ATLANTA, GEORGIA

A COLLECTION OF TECHNICAL PAPERS
ATLANTA, GEORGIA
SEPTEMBER 7-9, 1988

For permission to copy or republish, contact:
The American Institute of Aeronautics and Astronautics
370 L'Enfant Promenade, SW
Washington, DC 20024-2518

A COLLECTION OF TECHNICAL PAPERS

AIAA FLIGHT SIMULATION TECHNOLOGIES CONFERENCE

SEPTEMBER 7-9, 1988 /ATLANTA, GEORGIA

Printed in the U.S.A.

AIAA Flight Simulation Technologies Conference
September 7-9, 1988/Atlanta, GA

TABLE OF CONTENTS

Paper No. **Title and Author** **Page No.**

Session 1—Visual System 1

Session 2—Computation Methods

Session 3—Topics in Motion Cueing

Session 4—Training Systems

Session 5—Simulation Systems

Session 6—Combat Simulation Techniques

Session 7—In-Flight Simulation

Session 8—Simulation Applications

Session 9—Simulation Verfication, Validation and Testing

Session 10—Perceptual Aspects of Simulation

Session 11—Space Applications

88-4575-CP

WIDE FIELD OF VIEW HELMET MOUNTED DISPLAY SYSTEMS FOR HELICOPTER SIMULATION

Loran A. Haworth
and
Nancy M Bucher
U.S. Army Aeroflightdynamics Directorate
Ames Research Center, Moffett Field, California

Robert T. Hennessy
Monterey Technologies, Inc.
Carmel, California

Abstract

Simulation scientists continually pursue improved simulation technology that furthers the goal of closely replicating the "real world" physical environment. The presentation/display of visual information is one such area enjoying recent technical improvements that are fundamental to conducting simulated operations close to the terrain. Detailed and appropriate visual information is especially critical for Nap-Of-the-Earth (NOE) helicopter flight simulation where the pilot maintains an "eyes-out" orientation to avoid obstructions and terrain. This paper elaborates on visually-coupled Wide Field Of View Helmet Mounted Display (WFOVHMD) system technology as a viable visual presentation system for helicopter simulation. Critical research issues on helmet mounted displays are reviewed. Tradeoffs associated with this mode of presentation as well as research and training applications are discussed.

Visual Presentation History/Background

One of the persistent efforts in simulation research and development has been to increase the visual area of the simulated outside world. The ideal goal is to present a detailed scene over the entire area not occluded by the aircraft itself. Early real time visual scene presentation used limited Field-Of-View (FOV) (48 X 37 degree) flat screen projections of terrain model board images transmitted from closed circuit TV cameras to simulation projectors (fig 1). Use of CRT/beam splitter systems with terrain TV camera/model systems also presented only a limited FOV. As computer-generated-imagery (CGI) became available in the 1970's, a wider FOV was possible although early CGI scene content was of marginal realism. The CGI presentation/display devices included CRT/beam-splitter "windows" (fig 2) and domes (fig 3).

One long-recognized method for economizing on the hardware requirements for a large visual scene is to present imagery only in the portion of the field where an observer is actually looking called the Instantaneous Field Of View (IFOV) rather than fill the entire area where the observer might look, the Field Of Regard (FOR). Doing so requires monitoring the position of an observer's head, and perhaps the eyes, to determine what portion of the potential scene should be generated and displayed. Embodiments of this approach are know as visually-coupled systems. Kocian (ref 1) gives a general description of the advantages of these systems.

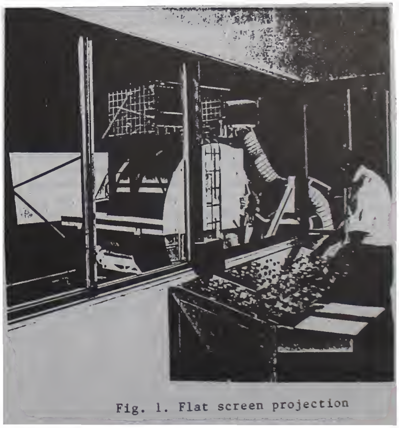


Fig. 1. Flat screen projection

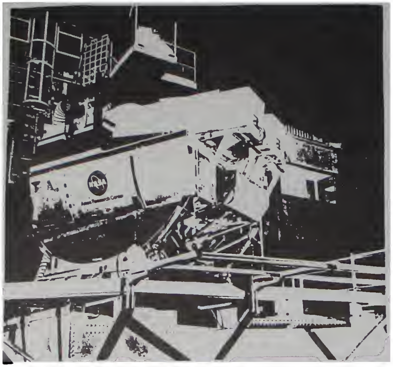


Fig. 2. CRT/collimation "Windows."

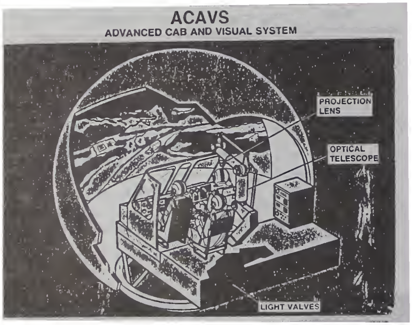


Fig. 3. dome

Dome presentation systems have the capability for visually coupled image presentation by using an area of interest coupled to the pilot's head movement, thus arriving at a much larger or detailed FOR. An early realization of a visually-coupled dome system, the Helmet-Mounted Laser Projector is discussed in reference two. In this device, a projector mounted on the pilot's helmet casts a scene on a dome screen. A fiber optic bundle conveys the imaging laser light to the projector. The part of the scene to be generated and projected is determined by tracking the pilot's head position. This system has evolved over the past several years. The latest version, incorporating an eye-tracked area of interest, is being tested in the Visual Technology Research Simulator at the Naval Training Systems Center, Orlando, Florida (ref 3). A visually-coupled area of interest dome system is also the subject of research at Williams Air Force Base, Arizona.

Conventional aircraft simulator displays which include both the dome projection and CRT/beam-splitter window technology normally require extensive housing requirements to support the weight and size of the assembled simulator. Producing enough light to cover a large area is also a problem. Most projection systems produce luminance levels between one and five foot-lamberts. A more detailed comparison of visual presentation systems for simulation can be found in reference 4.

Wide Field-Of-View Helmet Mounted Display (WFOVHMD)

Visually-coupled Helmet Mounted Displays (HMD) are a recent presentation method (fig 4) designed to provide the pilot with a broad range of visual information. The basic components of helmet mounted display are the helmet, display optics, electronics attached to the helmet and head tracking electronics. Helmet orientation and position are sensed by the head tracking electronics and sent to the image generation system. Appropriate graphic/visual imagery is then conveyed by an optical train to each eye. In the past HMD systems development has primarily been directed toward presenting sensor video imagery in flight vehicles with only recent application to simulation as the primary visual display system (refs 5, 6 and 7).

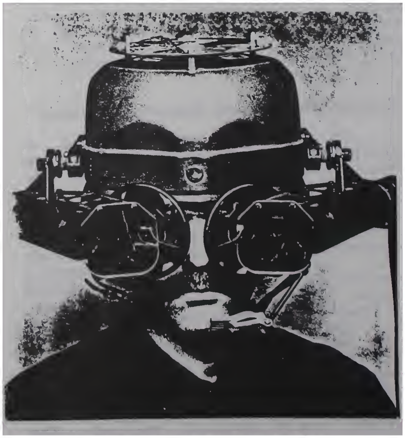


Fig. 4. Helmet Mounted Display

While the benefits of HMD's for simulation were recognized some time ago, it is probably the advantage of using HMDs in actual aircraft that is the main impetus for their development. Advanced military scout and attack helicopters are increasingly dependent on Forward Looking Infra-Red (FLIR) and other sensor imagery for piloting during night and adverse weather. However, the limited FOV of imaging sensors require that they be turned to the direction of interest; obviously, this is presently best accomplished by head movements.

HMDs in simulators offer unique advantages over projection and CRT array displays because they can be used as the primary display system for all imagery. Given a capable CGI and graphics visual system, a visually-coupled WFOVHMD provides an extensive spectrum of visual information that is competitive with real world viewing and beyond. The primary visual reference is calculated at the pilot's head position allowing a direct and more natural view of the simulated outside CGI scenery and inside the physical cockpit. The same display can present world-stabilized, representations of the flight environment and associated symbolic displays, mimicking a heads-up display system, for example, and head-stabilized displays such as symbology for weapons and threat. Graphics and symbolic information can also be mixed with the visual presentation to represent virtual cockpit displays and switches. Additionally, the pilot has the unparalleled capability of viewing his own simulated aircraft thus allowing him the important perception of physical helicopter dimensions and his attachment to an actual airframe.

The WFOVHMD with a large FOR presentation is especially important for simulation of future scout and attack helicopters that operate just a few feet from obstructions and ground. Future cockpit technology is moving toward greater reliance on synthetic imagery, multi-function pictorial and symbolic displays, night vision sensors and expert systems. The consequence is that the technology of the aircraft cockpit and the simulator are merging. The manifestation of this technological merging is the WFOVHMD. The primary benefits of WFOVHMD for both aircraft operations and simulation are unlimited fields of regard, compactness, potential availability of information including control and tailoring of the visual environment. Moreover, training in a simulator with a helmet-mounted display and computer generated imagery will now be a more realistic representation of the actual aircraft **environment** than has been possible previously.

CREW STATION RESEARCH AND DEVELOPMENT HELMET MOUNTED DISPLAY

One example of visually-coupled WFOVHMD simulation technology development is found in the Crew Station Research and Development Simulation Facility (CSRDF) located at the Ames Research Center in Mountain View, California (ref 8). The CSRDF facility was designed and funded by the U.S. Army, built by CAE Electronics, Ltd.,and managed by NASA-Ames for simulation research directed towards resolving critical mission equipment packages and pilot/aircraft interface issues for the next generation of scout/attack rotorcraft.

The crew station WFOVHMD system (figs 5 and 6) consists of a lightweight, custom-fitted helmet with two sets of helmet optics. Visual imagery which includes both color scene generation and flight/system symbology is transferred to the helmet optics via two fiber-optic bundles, from four high-brightness light valve projectors. A software mask is provided that blanks the video image surrounding the internal cockpit to allow concurrent viewing of the video imagery and cockpit. Helmet/head movements are monitored by an infrared head tracker using an LED array mounted on the helmet and sensors mounted in the overhead structure. An angular rate sensor package located on the back of the helmet provides lead predictions to compensate for visual imagery transport delays. The helmet serves as a stable platform to mount the displays for constant display orientation. The CSRDF fiber optic helmet mounted display system is unique in its capability to simulate a range of fields of view, up to 120 degrees horizontal and 67 degrees vertical. The range of the field of regard can also be varied up to a complete 360 degree surround. The Compuscene IV (CIV) computer generated imagery system provides the visual flight environment through which the pilot flies. The CIV is capable of simulating effects of FLIR sensor noise, resolution, automatic vs manual gain control, polarity reversal, blooming, temperature and time of day effects, and sensor position on the simulated aircraft.

One of the key technologies required for the next generation of scout/attack rotorcraft which has not yet been adequately specified in terms of the pilot interface requirements is the pilot's helmet mounted display. Because the helmet mounted display will provide the window to the visual world through which the pilot must fly, it is critical that this device be optimized in order to provide the maximum amount of information to the pilot to insure safety of flight and mission effectiveness. The overall visual

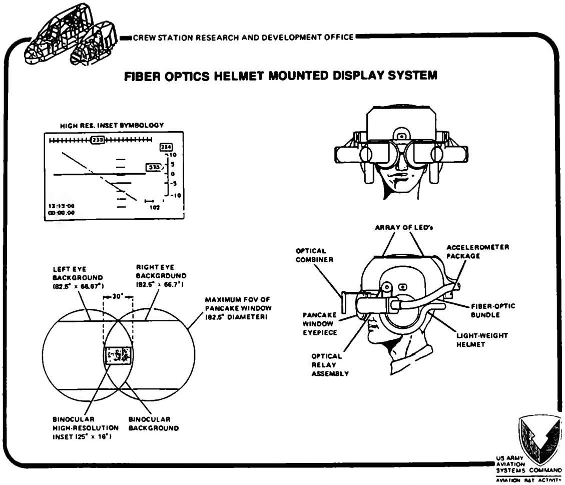

CREW STATION RESEARCH AND DEVELOPMENT OFFICE
FIBER OPTICS HELMET MOUNTED DISPLAY SYSTEM
HIGH RES. INSET SYMBOLOGY
LEFT EYE BACKGROUND (82.5° x 66.67°)
RIGHT EYE BACKGROUND (82.5° x 66.7°)
30°
MAXIMUM FOV OF PANCAKE WINDOW (82.5° DIAMETER)
BINOCULAR HIGH-RESOLUTION INSET (25° x 18°)
BINOCULAR BACKGROUND
ARRAY OF LED's
OPTICAL COMBINER
ACCELEROMETER PACKAGE
FIBER-OPTIC BUNDLE
PANCAKE WINDOW EYEPIECE
LIGHT-WEIGHT HELMET
OPTICAL RELAY ASSEMBLY
US ARMY AVIATION SYSTEMS COMMAND
AVIATION R&T ACTIVITY

Fig. 5. Crew Station WFOVHMD System

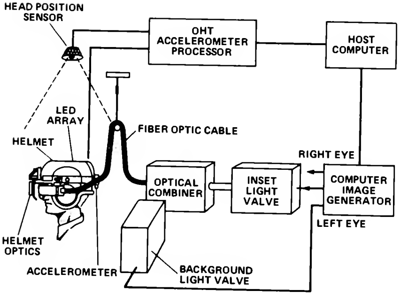


Fig. 6. Crew Station WFOVHMD System

display simulation capability of the CSRDF will allow a thorough examination of critical pilot interface requirements for visually coupled helmet mounted display systems.

There are a number of visual display parameters which must be examined in order to determine the optimal specifications for helmet mounted displays. These parameters are listed below:

- Instantaneous field of view (FOV)
- Field of regard
- Resolution
- Brightness
- Transmissivity
- Monocular, binocular,biocular presentation
- Pilot/sensor offsets

The determination of optimal helmet mounted display characteristics based on the parameters listed above requires a series of trade-off studies to be conducted both in part task and in full combat mission simulations. These studies include:

- Trade-off between field of view and resolution
- Monocular vs binocular viewing conditions
- Biocular vs binocular viewing conditions
- Image brightness vs display transmissivity

Simulations must also be conducted to examine the disorientation effects resulting from pilot/sensor offsets and image misalignment with true environmental spatial location.

Unique Simulation Capabilities Of Visually-Coupled WFOVHMD

General

Mission requirements will determine the design of future scout and attack helicopters for optimum outside viewing capability to support both NOE flight operations and the detection and countering of ground and air threats. To adequately model the visual capabilities of these cockpits, simulation of a large IFOV and FOR is mandatory. A visually-coupled WFOVHMD fills this need by allowing the pilot the ability to fly and to sight/track airborne and ground targets (fig 7) that are normally outside the visual envelope of both the dome and CRT window systems.

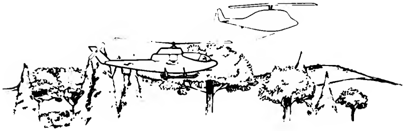


Fig. 7. Airborne target

Immediate Visual Environment

NOE operations often require close monitoring of objects and terrain in the immediate visual environment. Visually-coupled WFOVHMD permits the pilot to direct his visual environment to view objects immediately to the sides of the aircraft and vertically above and below. Maneuvers that have been visually difficult to accomplish such as a vertical re-mask into a "hover hole" after moving the aircraft laterally, can now be accomplished as easily as in the actual helicopter by viewing below and to the sides (fig 8) to clear the area under the aircraft prior to descent. The increased resolution of a WFOVHMD and the ability to view the immediate surrounding terrain in combination with the improvements in CGI micro-texturing complement each other. The three combinations permit the pilot to detect small precision hover displacements of a few inches rather than feet.

and the ability to view the immediate surrounding terrain in combination with the improvements in CGI micro-texturing complement each other. The three combinations permit the pilot to detect small precision hover displacements of a few inches rather than feet.

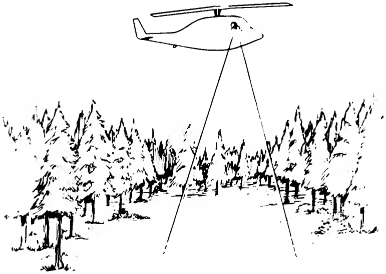


Fig. 8. Remask into "Hover hole"

This visual capability has implications for both training and for research. The frustration shared by many pilots trying to maintain a stable hover point without appropriate visual cues will decrease. The pilot will no longer have to fly "blindly" in lateral flight since he can turn his head in the appropriate direction for visual information. A simulation pilot can be trained closer to the same flight performance standards as expected in the aircraft and researchers developing flight control laws can now explore flight control law possibilities that were not practical due to the lack of visual feedback cues.

Target Presentation

Presentation of visual targets/objects has always been difficult in both dome and CRT based systems due to restricted FOV and FOR. Targets/objects must appear in areas aligned with the CRT windows or in the available dome projection area to be seen. To maintain visual contact, particularly with air targets, pilots often compensate for the lack of a simulated wide FOV and FOR by executing extreme attitude maneuvers, to keep the threat aircraft within the limited visual area. Head tracked WFOVHMD improves this situation by allowing the pilot to visually track the target anywhere within the simulated canopy envelope by turning his head without the need to redirect his own simulated aircraft. Consequently, when a pilot maneuvers the aircraft in simulation using the WFOVHMD, the maneuvers are more likely to be typical of actual flight. For example, the pilot can now view areas above and to the rear of the aircraft that are normally simulator blind areas.

The WFOVHMD allows for more effective transfer of training between the simulator and actual aircraft since a pilot does not have to unlearn visual search tactics formed in an actual aircraft to compensate for limitations in a particular simulator. The head tracked WFOVHMD visual system also has implications for engineering researchers trying to monitor the envelope of aircraft during targeting since the motivation to exceed the helicopter sideslip envelope to visually acquire the target will decrease.

Simulation of Airborne Visually-Coupled Sensor Systems

Future attack and scout helicopters are projected to incorporate sensor systems that are coupled to the pilot's head movements for operations requiring night vision pilotage capabilities (fig 9). Systems of this nature can be found on the Army's newest attack helicopter, the AH-64 Apache. Simulation of head directed pilotage sensors are possible with WFOVHMD by modification of the CGI visual table and calculation of a displaced eye point to replicate the sensor view point on the aircraft such as the front or mast of the helicopter (fig 10). Characteristics of the visual chain can also be modeled to simulate the FOV and other attributes of actual aircraft HMD such as monocular or binocular optic displays, and head tracking properties. For example, display of stereoscopic information with independent eye channels and the value of stereopsis in various flight and visual search tasks can be investigated. Interactions among different image sources such as FLIR, TV and image intensifiers can also be explored.

Head Directed Aiming

Advanced military helicopters will use head directed weaponeering during daylight and night (sensor) flight in the combat environment. Simulation investigations of aircraft targeting system methods of designation and targeting, would be difficult without a HMD for head directed sensor visual and targeting/designation symbology. Head tracking systems also provides the researcher with an additional potential measure of performance. For example, head movement data could be used to evaluate changes in pilot scan patterns as a function of the FOV, other characteristics of sensor imagery, flight tasks, and experience. Figure 11 demonstrates the simulated use of head-directed weapons during flight.

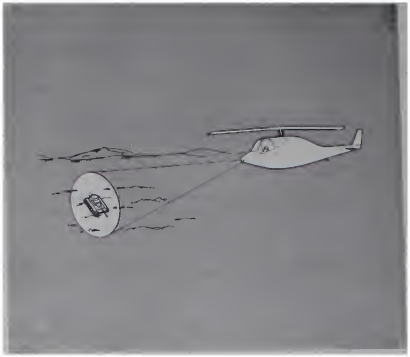


Fig. 9. Night Vision Pilotage

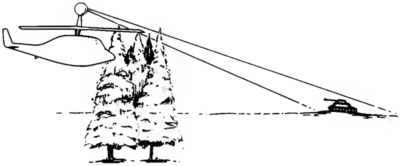


Fig. 10. Sensor View Point

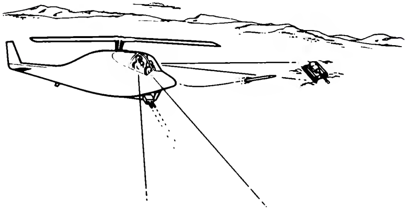


Fig. 11. Head Directed Weapons

HMD Symbology

Many types of symbolic presentations are possible to aid in performing various flight tasks to include projected navigation path ways in the sky. Helmet mounted display systems (fig 12) naturally lead to research on appropriate symbology for flight, targeting, designation, system cuing, and sensor display images. Other research needs include establishing appropriate symbolic and pictorial display representations for HMDs and determining appropriate frames of reference for symbol and image stabilization.

Symbolic virtual images are possible with an HMD that can project cockpit systems, switches and functions without actual hardware. A virtual cockpit display can be created and made to appear to be located on a blank panel space or on the windscreen as an example. Thus, panel display designs can be manipulated and evaluated by the same process used to alter HMD displays.

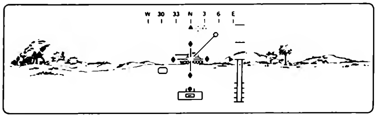


Fig. 12. Flight Symbology

Visual Aircraft Dimensions

During NOE flight the pilot often has difficulty estimating the dimensions of the simulated aircraft for safe pilotage near obstructions. This is especially true as the technical capability to produce realistic and visually dense data base objects such as trees has increased in response to the Army's need to simulate flight in the low level environment. With visually-coupled WFOVHMD and improved CGI systems the pilot operator can potentially view his own aircraft structures. This has a great impact on the way the pilot operates the aircraft since he can now see physical clearances. Other benefits include visual verification of aircraft configuration changes such as lowering of aircraft gear and tilt of the nacelle on a tilt rotor aircraft. In the area of emergency procedures the pilot can now realistically perform such tasks as visually verifying aircraft fires (fig 13) and monitoring combat damage to the airframe. This capability allows for more realistic depiction of real world visual events.

Simulation Structural Requirements

Both CRT window and dome based simulators require extensive housing requirements and are very heavy. Some airframers have been forced to expand simulation buildings to accommodate the domes. The size and weight of the CRT window and dome systems often limit the ability to move these massive systems on motion bases without relatively powerful motion systems. Helmet mounted displays have the potential for greatly reducing housing and motion base requirements since the primary display device is attached to the pilots head. Reduced size and weight of the simulator implies better motion simulation for the increasingly agile and maneuverable helicopters of the future.

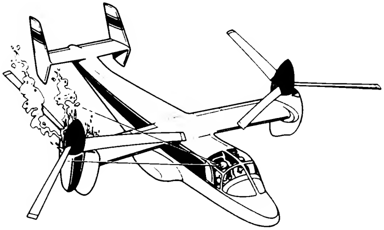


Fig. 13. Visual Verification

Simulator, Aircraft and Embedded Training Applications

The helmet mounted display system can provide a visual display for both the simulator and aircraft if the display accepts video signals from the simulator CGI and the aircraft sensor system. Since future aircraft may be primarily controlled by electronic signals, embedded training for the pilot in his own cockpit could be performed with relative ease by simple attachment through an umbilical cord to a simulation van. Electronic control signals would be interpreted at the simulation van computer and appropriate electronic outputs would be reflected on ship board displays and video signals to the pilot's HMD (fig 14). At this point networking with other sister aircraft would be possible thus allowing flight section members the capability of operating their own aircraft while interacting with other team members in support of future missions that could include the latest threat projections.

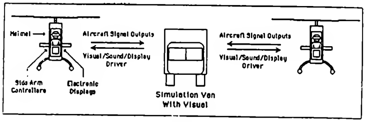


Fig. 14. Example of Embedded Training using Advanced Flight Vehicle

Advantages and Disadvantages

Several unique advantages are gained for researchers and trainers by use of helmet mounted displays. In general, wide FOV and unlimited FOR at a high brightness are the most important improvements gained with the helmet mounted display. Not only is the HMD compatible with visually-coupled pilotage sensor systems but also provides the capability to conduct off-axis targeting and designation for military helicopters. Other advantages include the use of software graphics for virtual cockpit images, viewing aircraft structure and the ability to package a visual system into a lighter, more compact simulation facility.

Disadvantages associated with HMD are similar to those found with visually coupled systems in aircraft. The helmet has to be properly fitted to the individual pilot and proper optical alignment must be established. The weight, center-of-gravity, and inertia must be appropriately considered and the HMD normally requires more set-up time for adjustment. The ability to conduct simulation demonstrations are somewhat more difficult due to the time require for proper HMD setup.

Technology Outlook

Many ongoing research and development efforts are underway to reduce the technical shortcomings. These efforts include reducing the weight and inertia of the helmet, adjusting the center of gravity of the helmet, and reducing the weight and size of the system package. Studies are now being conducted to develop a helmet system for use in large amplitude motion base systems such as the Vertical Motion Simulator located at NASA-Ames Research Center. When completed this motion worthy HMD will a valuable research and design tool for for flight-worthy, visually-coupled, HMD systems.

REFERENCES

1. Kocian, Dean F., "A visually - Coupled Airborne Systems Simulator (VCASS) - An approach to visual Simulation." Proceedings of the 1977 Image Conference, Williams Airforce Base, Arizona, 17-18 May 1977.

2. Breglia, Denis R., Michael Spooner, Lobb, Dan, "Helmet Mounted Laser Projector." Presented to the 1981 Image Generation Display Conference II, Scottsdale, Arizona, 10-12 June 1981.

3. Hettinger, Lawrence J., Berbaum, Kevin S., Kennedy, Robert S., and Westra, Daniel P. "Human Performance Issues in the Evaluation of a Helmet-Mounted Area-of-Interest Project." Proceedings of the 1987 Image Conference IV, Phoenix, Arizona, 23-26 June, 1987.

4. Cook, Anthony M. "The Helmet-Mounted Visual System in Flight Simulation." Presented to the 1988 Royal Aeronautical Society Conference, London, England, 12-13 April, 1988.

5. Kocian, Dean F., "Virtual Panoramic Display (VPD) Helmet Systems." AAMRL/HEA, Visual Display System Branch, Wright-Patterson AFB, Ohio 45433-6573, not dated.

6. Gunther, Marty, "Advanced Helmet Integrated Display Systems - A New Concept for The Future." Presented to the 1987 Technical meeting of the Anionics Section Air Armament Division FT Eustis, Virginia, 18-19 Nov. 1987.

7. Fisher, Robert A., Tong, H.M., "A Full-Field-of-View Dcme Visual Display for Tactical Combat Training." Singer Link Flight Simulation Division, Presented to the 1987 Image Conference IV, Phoenix, Arizona, 23-26 June 1987.

8. Lypaczewski, P.A., Jones, A.D., and Voorhees, J.W., "Simulation of an Advanced Scout Attack Helicopter for Crew Station Studies." Presented at the Royal Aeronautical Society International Conference on the Acquisition and use of Flight Simulation Technology in Aviation Training, London, England, April 1987.

PROCESSING PSEUDO SYNTHETIC APERTURE RADAR IMAGES FROM VISUAL TERRAIN DATA

M. E. Sturgell
Avionics Laboratory
J. R. Lewonski
Flight Dynamics Laboratory
Wright-Pat AFB, Ohio

Abstract

The paper discusses a solution in the area of simulating Synthetic Aperture Radar (SAR) High Resolution Map imagery from Terrain Board visual data to aid a pilot in determining safe landing vectors on bomb-damaged runways. The Visual data was obtained from a Terrain Board system using a method that considered image centers, rotations, and fields-of-view. The areas of concern were obliquely lit during the photographing stage to create the necessary shadowing characteristic of SAR. The image processing software was developed to convert the collected still images into pseudo-SAR images. An image processing tool was written and used to support the development of this image conversion algorithm. Another program was written to process and store the large number of pseudo-SAR images onto a videodisc as an ordered library. This library was randomly accessed and displayed in a real-time flight simulation through a videodisc based system. After the pilot designated a reasonable touch-down point on the SAR cockpit display the simulation host passed the appropriate coordinates to the Inertial Navigation Systems which computed the proper approach vector for display.

Background

Program Objective

The origin of this study was a direct response to the requirements of the Air Force Short Take-off and Landing STOL, STOL/Mission Technology Demonstrator (S/MTD) Program. The objective of this program is to provide the Air Force with the simulated capability to land and takeoff on a bomb-damaged runway with a Minimal Operating Strip (MOS) without degrading the air-to-air capability and performance of the baseline aircraft. The research effort of the Air Force directly supports the Advanced Development Program Office in the validation of contractor development in Control Law design, Pilot Vehicle Iterface symbology, safety of flight, and evaluation of autonomous landing tasks.

Approach

To effectively perform the defined tasks for the Advanced Development Program Office the following simulation requirements were established for proper evaluation of the data. A six degree-of-freedom motion base simulator, a SAR Sensor capability, and an advanced Pilot Vehicle Interface capability were essential for the validation of data. The complete feel-system of the LAMARS (Large Amplitude Multi-Mode Aerospace Research Simulator) of the Flight Dynamics Laboratory's Simulation Facility was chosen as the simulation test-bed for the evaluation of the contractor's efforts.

To check the compiled data the LAMARS cockpit controls and output displays were reconfigured and modified to provide a realistic environment to the pilots who were critiquing the system modifications. For the performance evaluation of the autonomous landing tasks, a real-time SAR Sensor capability had to be developed and implemented to enable the pilot to view the bomb-damaged runway and acquire an adequate MOS. The requirements, limitations, approach, and evaluation of a SAR Sensor capability for the STOL studies involving bomb-damaged runways are discussed extensively. The development of a facility SAR Sensor capability in support of real-time simulation efforts is the specific area of concern which this paper addresses.

Real-time SAR Sensor Simulation

Requirements

The development of a SAR Sensor capability for the simulation facility was identified as a major concern due to its role in the program's autonomous landing tasks. The initial requirement was the simulation of an APG-70 Radar for the 5000 to 1 Terrain Board bomb-damaged runway. Further studies were conducted to define the specific requirements for the completion of the simulated landing tasks utilizing a SAR Sensor. From the evaluation of the landing tasks, it was determined that only 2 out of the 5 APG-70 Radar modes were necessary for mission assessment. These map modes are the High Resolution Map (HRM), which provides 8.5 ft resolution maps at close ranges and 17 ft resolution maps at more distant ranges, and the Precision Velocity Update to correct for the errors between the radar and the Inertial Navigation System (INS). The defined task of this project was to supply a simulated SAR Sensor HRM display capability for the designated areas of the Terrain Board.

Alternatives

The feasibility of utilizing a Radar Landmass Database System (Defense Mapping Agency Data) or Off-the-Shelf Radar Simulation Systems was evaluated according to the SAR Sensor guidelines established. Due to the limitations of the facility's terrain board data, the limited radar modes required, and the inherent high-costs of these Radar Simulation systems this approach was determined not to be the optimal solution. Instead an In-House effort to the development of the required SAR Sensor capability was determined to be the most beneficial and feasible solution.

The advantages of an In-House capability buildup allowed for the tailoring of the task to the desired performance specifications, the joint effort of both the Avionics Laboratory (Sensor Expertise) and the Flight Dynamics Laboratory (Simulation Facility) define and develop the task implementation methodology, and provided the most cost effective approach to satisfying the defined simulated SAR Sensor criterion.

Implementation

Real-time Considerations

The basic task of producing a simulated real-time APG-70 SAR Sensor HRM capability involves producing the SAR image within three seconds of pilot command. Ideally, a system which could derive simulated SAR images from existing visual data in real-time would best fit the requirement. However, the amount of data processing required to perform such a conversion even with today's real-time image processing systems precludes this from being a viable approach. Alternatively, simulated or pseudo-Sar images could be generated from visual data in non-real time and then stored in a random access library for later real-time retrieval. Using videodiscs as a random-access storage media for such a library provides both the necessary real-time operation and cost effectiveness desired.

Following this approach, the next problem to address is a means of collecting, processing and storing the images onto videodisc in an order amenable to quick retrieval. To provide an image database for the entire gaming area for all possible headings and altitudes would present us with an unmanageably large number of still images to deal with. Luckily, the scope of the effort only requires SAR images from a small geographical area, specifically a bomb damaged runway, and the number of discrete images required are therefore significantly reduced. By choosing an image overlap factor of 95% horizontal and 95% vertical, limiting radar headings to 2 degree increments, and limiting altitude variations to two discrete values based on the HRM resolutions (8.5 and 17 ft. resolution), the number of images required is further reduced while still maintaining enough flexibility to provide the accuracy required to accomplish the task.

Data Gathering

Data gathering was implemented by recording discrete video still images of the subject terrain onto a write-once, read-many (WORM) type videodisc. To simulate the bomb damage for the runway, a magnetic overlay was created depicting the damage in 3D relief. By using an oblique lighting source mounted to a ring on the video probe such that the light always appeared to be radiating from the bottom of the video image toward the top, the SAR shadowing effect was approximated.

One problem with the resultant images was that the intensity of the oblique lighting required to produce definite relief shadows also created a definite hot spot toward the bottom of the image. This problem was remedied by videographing a featureless black-matte image for each of the two altitude groups, and later summing it's inverse with the original images.

Accuracy of registration of the video stills would be critical to the usability of the resultant simulated SAR images for two reasons. First, the pilot would be required to mentally correlate the SAR display with the out-the-window scene. Second, specific points on the SAR display would have to correlate closely to actual ground coordinates so that the pilot could identify and designate a safe touchdown point(s) and landing vector(s) which could then be passed to the INS for a precision landing approach. Fortunately, this close correlation was made easy since the same video probe used for the visuals display within LAMARS was used to take the video stills that would be processed into the SAR images. Positioning and angular registration was therefore identical. The only loss of accuracy in fact was as a result of limiting the number of pseudo-SAR images as mentioned above, and those spatial limitations proved to be within the bounds of usability for the purposes of the study.

Data Manipulation

Algorithm Development:

Once the obliquely illuminated video stills were recorded onto the WORM videodisc, they had to be further processed to look like realistic SAR returns. This was accomplished in the digital domain by utilizing image processing equipment. Both because of cost considerations and flexibility of use, the Data Translations DT-2858/2851 card set for the IBM PC was selected as the image processing equipment of choice. This card set is supported by an extensive library written for Microsoft C. This made the development of the specialized software required for the image processing

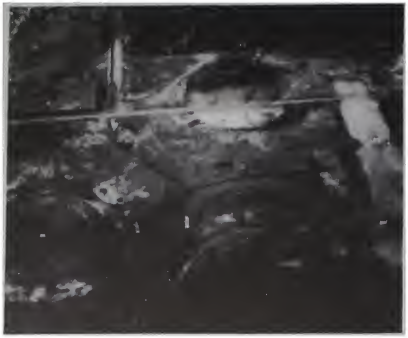


Fig. 1 - Out-the-Window Cockpit Area Photograph.

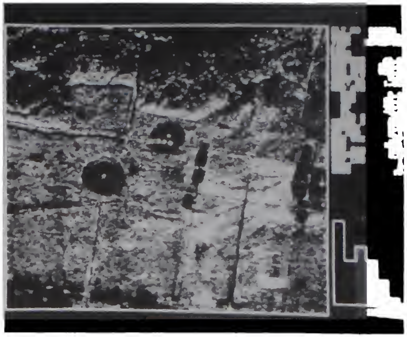


Fig. 2. - SAR Photography of the Same Area.

task relatively easy to derive in-house. First, a general purpose image manipulation program was written to facilitate the heuristic experimentation required to develop the visual-to-SAR image processing algorithm. This program, DT-TOOL, was written to take advantage of all of the card set's image processing features which include a wide variety of arithmetic, logical, and spatial manipulation capabilities.

In order to manipulate the visual images into pseudo-SAR images, we first had to define the distinguishing physical characteristics which separated them. With a visual image, light illuminates the scene from all angles in a non-coherent fashion. The portion of this non-coherent light within the eyes' field of view is perceived from the eyes' position as a scene with a specific orientation and distance. Further, because of the stereoscopic nature of the human eyes, depth is perceived.

With SAR, as with any radar image, many of these factors are either simplified or eliminated due to the nature of the sensor itself. First, radar emissions are coherent in nature, with the source of "illumination" and the position of "perception" being largely the same. Because of this coherency, objects which are more normal to the radar emission vector will tend to return more energy, and therefore appear "more illuminated" than terrain or objects which are more parallel to the emission vector. For this reason, true SAR images tend to show glint for vertical surfaces facing toward the aircraft, and deep shadows behind the opposite surfaces. Second, because radar "illumination" is, in fact, electromagnetic in nature, denser, more metallic surfaces will tend to reflect the energy more efficiently than less dense, non-metallic surfaces much the same as lighter, "more white" objects tend to return more light energy to the eye than do darker, "more black" objects in the visual world. Because there is seldom a direct correlation between the visual color of objects and their density or metallic composition, it would not be desirable to key on color for a programmatic visual-to-SAR conversion algorithm. For this reason, it is preferable to simplify the problem by starting with black and white visual data. Third, since there is a single point of emission for the radar energy, there is no perception of depth requirement for the SAR image. Therefore, the 2D nature of the original video still image works in our favor automatically. This situation is further simplified when the specific operational characteristics of SAR are considered. The ability to produce ground mapping data from radar relies upon the precise timing of emitted and returned radar pulses while the aircraft travels over a known distance, thus creating the synthetic aperture for which this type of radar gets its name. Pulses are transmitted and their reflections received in timed groups. Within each timed group, the emitter-to-ground range is varied through slight angular steps in the direction of emission. These intragroup returns form a narrow range strip, or cell, of reflectivity data. By repeating this process over the length of the synthetic aperture and combining these cells side-by-side, a two dimensional composite of the scanned ground area is produced. This resultant image appears the same as if the area were viewed from directly overhead. By taking the video still images with a 90 degree look down angle, the same effect can be achieved.

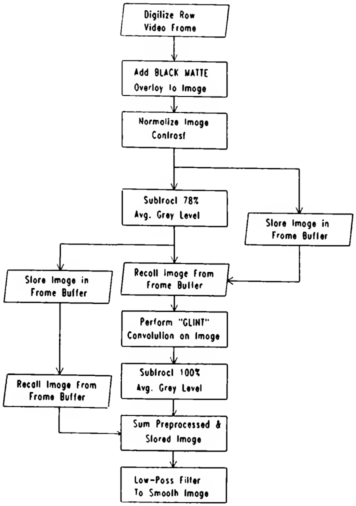
Figures 1 and 2 graphically demonstrate the differences between visual and SAR representations of terrain and objects. Figure 1 is a photograph of a patch of ground as viewed from the cockpit window. Figure 2 is this same patch of ground as seen by an actual SAR. Notice how acutely the three trucks in the upper circle are detected by the SAR. Also notice the glint associated with the boundary between field and forest, as well as the dark patches representing extremely flat ground and water areas.

Algorithm Implementation:

Since there were a large number of images to be processed into pseudo-SAR images, it was highly desirable to define an algorithm which could automatically and consistently produce the same results without human intervention. Keeping this constraint in mind, the following algorithm was devised using the developed user-friendly DT-TOOL software. Starting with the visual stills as described above, pseudo-SAR images were created by first digitizing the video still. Next, the previously digitized black-matte was added to this image to minimize hot-spotting caused by the oblique lighting during videography of the original images. To insure consistency across a large number of photographs, the resultant image was contrast normalized by redistributing pixel grey values according to the average of the four most frequently occurring grey values in the un-normalized image. This normalized image was then temporarily stored in a frame buffer for later re-use. Next, the same normalized image was darkened by 78% of it's average grey level in order to bias the darkest features to pure black. This darkened image was saved in another buffer and the previously stored normalized image was retrieved. This retrieved image was low-pass filtered to remove ambient noise from previous manipulations. To this filtered image, simulated glint was added by performing a directionally biased laplacian convolution to each pixel in the digitized image. Since we assume that the SAR emission is always oriented from bottom to top in our images, it is easy to likewise uniformly apply this convolution such that only the lower edges of grey-shaded boundaries within the visual image are enhanced. A five-by-five matrix as shown below was used to accomplish this convolution:

$$\begin{array}{rrrrr} -1 & -1 & -1 & -1 & -1 \\ 1 & -1 & -1 & -1 & 1 \\ 1 & -1 & -1 & -1 & 1 \\ 1 & 3 & 4 & 3 & 1 \\ -1 & -1 & -1 & -1 & -1 \end{array}$$

The resultant convolved image was also darkened by biasing it by its average grey level. This was done both to reduce noise introduced into the image from the convolution operation and to allow it to be added to the previously darkened image without causing grey value overflow. The next step involved the summing of this processed image to the darkened image retrieved from the frame buffer. The final step in the algorithm was to low pass filter this resultant summed image with a 3x3 matrix to smooth the edges and blur its overall appearance to produce the pseudo-SAR image. The entire operation is summarized below in algorithm form:


Digitize Raw Video Frame
Add BLACK MATTE Overlay to Image
Normalize Image Contrast
Subtract 78% Avg. Grey Level
Store Image in Frame Buffer
Store Image in Frame Buffer
Recall Image From Frame Buffer
Perform "GLINT" Convolution on Image
Subtract 100% Avg. Grey Level
Recall Image From Frame Buffer
Sum Preprocessed & Stored Image
Low-Pass Filter To Smooth Image

VISUAL TO PSEUDO-SAR IMAGE
PROCESSING ALGORITHM:

1. Digitize Raw Library Frame from Video Input.
2. Preprocess Raw Images:
   a. Add BLACK MATTE to Original Image.
   b. Normalize Image Contrast Ratio
      1. Determine average # of 4 most frequently accurring pixel values in the image.
      2. Use this "Hi-4" average to compute a "Homogeneity Ratio".
      3. Use this "Homogenity Ratio" to find scaled Max and Min Grey Levels. (This is what allows the algorithm to uniformly expand contrast levels of Raw Images with a wide variance in contrast ratios.)
      4. Expand the contrast ratio of the image by creating a pixel value lookup table that has Min and Max Grey Levels of the original image set to 0 and 255 respectively.
      5. Feed back the original image through this lookup table to create the contrast enhanced (normalized) image.
3. Save a Copy of the Preprocessed Image in a Temporary Buffer.
4. Subtract 78% of Preprocessed Image's Average Grey Level. (Drives all runway pixels to "very black".)
5. Swap Biased Image (step 4) with saved Preprocessed Image.
6. Perform "glint" producing convolution on Preprocessed Image.
7. Subtract 100% of resulting Average Grey Level to reduce noise.
8. Combine "glint" Image with previously saved "black runway" image (from step 4).
9. Smooth the Resultant Image with a 3X3 Low-pass Filter.

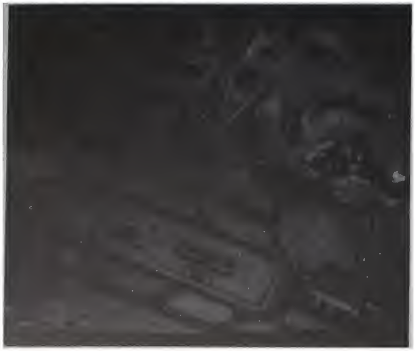


Fig. 3 - Preprocessed Video Image.

Figures 3 and 4 illustrate the effect of the visual to pseudo-SAR processing operation. Figure 3 is a photograph of one of the source frames. Figure 4 is the same image after processing.

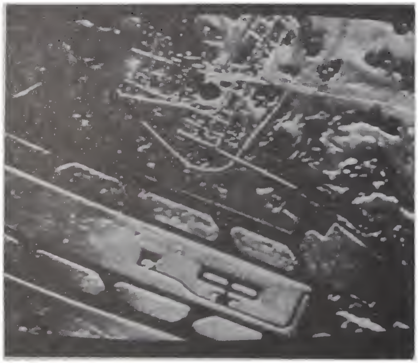


Fig. 4 - Same Image After pseudo-SAR Process Implemented.

SAR Library Storage

Once the algorithm was developed and tested, a second program, Sar Library Creator, (SLIC) was written which would implement it in an automatic fashion. Since the original video images were recorded onto videodisc, and both a videodisc player and recorder were available, it was a relatively simple matter to create a program which would select successive frames from the source videodisc (on the player), process the image according to the algorithm, and store the resultant image onto a new WORM videodisc using the recorder. Since each frame on both the source and destination videodiscs could be uniquely identified and accessed by frame numbers, SLIC was written to be table driven. This provided two desirable advantages. First, the source videodisc need not have consecutively perfect input frames, which allowed the program to skip over "out-takes" recorded during the initial videographing process. Second, by table driving the output to the recorder, any arbitrary storage scheme could be implemented as desired. This provided an easy method for optimizing the organization of processed frames for later real-time retrieval during the simulation.

The format used for the table generation of the pseudo-SAR Library locations was derived from parameters available on the real-time simulation computers. The image frame library addresses consisted of 5 digits (Number of digits based on the limited number of frames contained on the WORM videodisc.) with each digit having a significant parameter associated with it. The radar azimuth angle was rounded to the nearest 2 degree increment which established the value of the two least significant digits. Latitude and longitude data was used to generate a corresponding touchdown point value for the next two digits. The final digit was based on the range of the aircraft's position from the bomb-damaged runway. A simple algorithm to compute the desired image address was embedded in the simulation software and the calculated library location was sent to the videodisc player via RS232.

Real-time Display

The Real-Time display involved the integration of the cockpit controls, SAR Library Images, and radar symbology overlays. The following explanation pertains to the role pilot inputs play in the cockpit for the task performance of MOS runway designation. At large range values only the 17 ft. resolution (Figure 4) SAR maps could be displayed to aid the pilot in the location of a MOS runway clear of bomb damage. After locating a clear runway strip and the computed range of the aircraft from the runway is within the defined close range, the 8.5 ft resolution (Figure 5) SAR maps are utilized to support the pilot's designation of the touchdown point for the STOL aircraft. After the pilot has designated this touchdown point the flight computer feeds heading information and MOS location, based on the INS, for display on the HUD (Heads-Up Display) to aid the pilot in performance of the autonomous landing tasks.

The actual integration of the pseudo-SAR HRM mode is demonstrated in Figure 6. The pilot inputs are processed and sent to the Real-time Simulation Computer Network

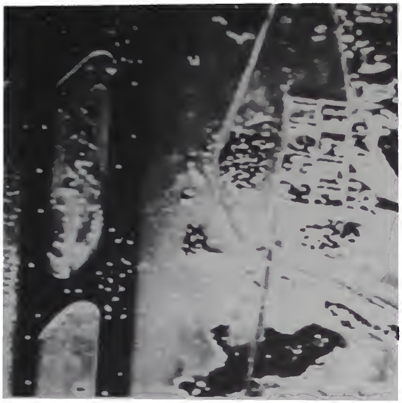


Fig. 5 - 8.5 ft Resolution SAR Map for Touchdown Designation.

which then begins an internal programmable clock for the SAR processing time constraint of 3 seconds. Based on the aircraft's range, azimuth, latitude, and longitude information at time of MOS designation the pseudo-SAR Image address is calculated and transmitted via RS232 to the Optical Videodisc Player containing the SAR Image Library. The video output of the Optical Videodisc Player is sent to a Digital Framestorer, which waits for a +5 v. discrete signal from the simulation computer at the end of the 3 second cycle clock, to update the requested SAR Image display. The SAR display is sent to a graphics station which then mixes the raster image with a stroke radar overlay for display on one of the simulator's MPDs (Multi-Purpose Display).

Conclusion

Technical Merits

The task of producing a realistic HRM SAR Sensor display capability increased the number of programs which the simulation facility could support. The In-House build-up of this capability increased the knowledge and aided in establishing a method of integrating the separate specializations of the individual laboratories. A facility format was determined for the retrieval of video data depicting accurate image perspective and shading for future processing into SAR imagery. The image processing method was automated allowing for the consistant processing and storage of the video images in the defined format. The pursuit in the development of the SAR capability enabled the facility to obtain a working knowledge of the operation of SAR which can be effectively applied to future radar simulations.

Effectiveness

Methodology:

A flexible methodology was established for the development and implementation of the simulated SAR capability, enabling individual programs to process and store tailored SAR Support Libraries. The derived methodology allows for integration of individual program inputs for the purpose of producing the desired SAR imagery perspective and location. Furthermore, the displayed SAR image was properly correlated with the simulator's out-the-window display. This made a more realistic environment for the pilot, who was responsible for accurately critiquing the system modifications.

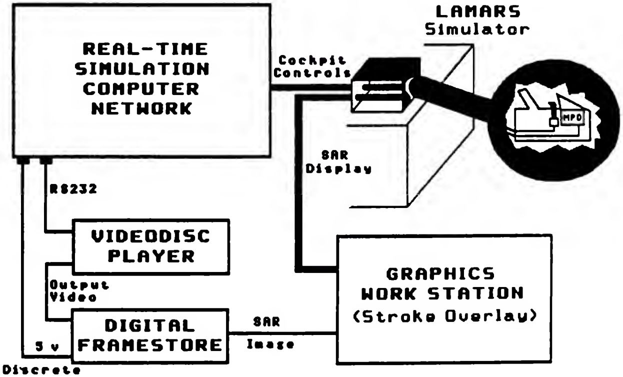


Fig. 6 - SAR Display Integration for Simulation Facility.

Cost:

The initial cost and the data requirements of a Digital Radar Landmass System were excessive when compared to the budget of the program. An alternative method of simulating SAR images from the facility visual data was developed and implemented to reduce the amount of required funds. The cost of this task included the purchase of a Videodisc Player and Recorder, fabricated 3-D relief bomb-damaged runway overlays, PC-based video image processing card set (Data Translations DT-2858/2851), board support software (DT-IRIS), and miscellaneous support equipment (i.e. Light source, WORM videodiscs, etc...). A more efficient use of funding is demonstrated for task performance when comparing these expenditures to the high cost of a Radar Simulation System. The following list is a cost breakdown of the individual items which supported the development and implementation of a SAR Sensor capability for display in the flight simulator.

Equipment	Cost
Videodisc Recorder	$ 22,750
Videodisc Player	3,645
DT-2828/2851 Card Set	4,930
DT-IRIS Software/License	1,520
Runway Overlay Fabrication	3,835
Support Equipment	1,400
Total:	$ 38,080*

(* Note: The total does not reflect the cost of 1.5 manyrs expended during the development and implementation of the process.)

Future Alternatives

Future alternatives for the simulation of SAR imagery for the facility can be separated into two viewpoints. The first point considers the present methodology implemented. With the inherent increase of technological development and available upgrades the processing method could be upgraded to handle the defined time constraint which would eliminate the need and cost of the storage media (WORM videodiscs). Tandem video probes would have to be introduced to provide the necessary terrain visual data required of the SAR Imagery and the out-the-window simulator display requirements. This method would increase flexibility by supplying actual terrain visual data as determined by the simulation computers for the processing of the required SAR Imagery real-time.

The second point involves the introduction of a Computer Image Generation (CIG) System for the replacement of the terrain boards. The cost of a CIG System is high but it enables the integration of multiple viewpoints, sun angle shading, increased image resolution, and larger gaming areas to be considered for out-the-window simulation display. The implementation of a SAR Sensor capability would involve the interfacing of another system to process the common database of the CIG System or the displaying of a preprocessed common SAR database taking into account the shadowing, color, and glint characteristics of SAR. Necessary future facility upgrades, program description, and technological developments have made this point a viable approach to future SAR Sensor simulation.

88-4577-CP

IMAGE EXTRAPOLATION FOR FLIGHT SIMULATOR VISUAL SYSTEMS

Keith Blanton
IVEX Corporation
4357J Park Drive
Norcross, Georgia 30093

ABSTRACT

Flight simulator visual systems which can render detailed terrain databases with realistic texturing are typically very expensive. However, constraining the terrain to be an absolutely flat plane can offer tremendous advantages in many important simulation scenarios over conventional systems. This paper outlines the fundamental principles behind a new approach based on this assumption and describes some of the implementation issues which must be considered. The result is a visual system which can generate images with high quality texturing and detail and maintain a guaranteed frame rate. These techniques are the basis around which the IVEX Corporation VDS-1000 flight simulator visual system was designed.

INTRODUCTION

Image extrapolation is used to create a displayed image from a prestored panoramic image called the keypoint image. The keypoint images are created off-line and stored on a high-capacity device, such as an optical disc. Essentially, the eyepoint position and attitude angles define, frame-by-frame, a transformation from the displayed image to the keypoint image. The keypoint image is sampled as the display is raster scanned, with the transformation defining the sampling grid. The requirement of flat terrain simplifies the transformation calculations and eliminates occultation problems which would be present if raised terrain were permitted. Aliasing is also more easily handled in the flat terrain case. Figures 1 and 7 are photographs of scenes generated in real time by a VDS-1000 using the described image extrapolation process.

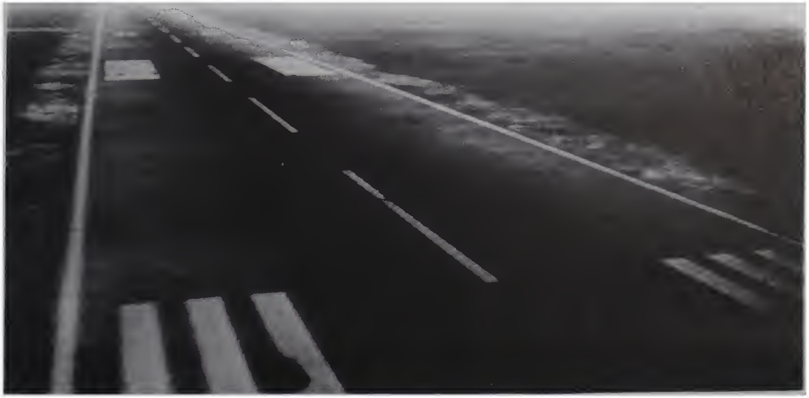


Figure 1: Runway Detail and Texturing

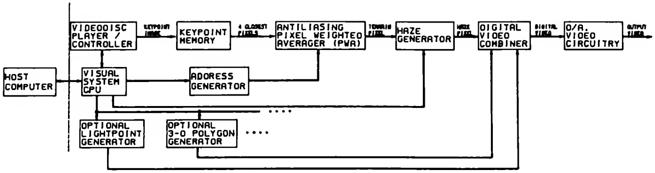


Figure 2: System Block Diagram

SYSTEM DESCRIPTION

Figure 2 is a block diagram showing the major functional components of the visual system. The host communicates position and attitude updates to the visual system CPU, which processes this information and conveys it to other visual system elements as needed. The CPU determines the best keypoint to use based on the eyepoint position and attitude and requests that the videodisc controller load this keypoint into the keypoint memory. A double buffer arrangement is used to allow one keypoint to load while another is used for scene generation.

Nine parameters, computed from the eyepoint position and attitude angles, are passed to the address generator at frame rate. These values are sufficient to specify how the display screen pixels and scan lines map into the panoramic keypoint image. Thus, the displayed image is derived from the keypoint image through a sampling process. The address generator computes an address into the keypoint memory for each pixel in the displayed image. This address has an integer and a fractional part for both the azimuthal and the elevational components. The four keypoint pixels surrounding this address are then accessed and weighted according to the fractional address parts by the antialiasing pixel weighted averager (PWA).

The output of the PWA can be viewed as filtered (antialiased) samples of the keypoint image. Now the haze generator blends each pixel produced by the PWA with the user-specified haze color based on the distance between the eyepoint and the point on the ground represented by that pixel. This "hazed" pixel can be combined with pixels computed by other graphics subsystems such as lightpoint and polygon generators. Since the terrain is perfectly flat, any object on or above the ground plane will occult the terrain, making the combining straightforward. Finally, the resulting digital image is converted to a video signal for display.

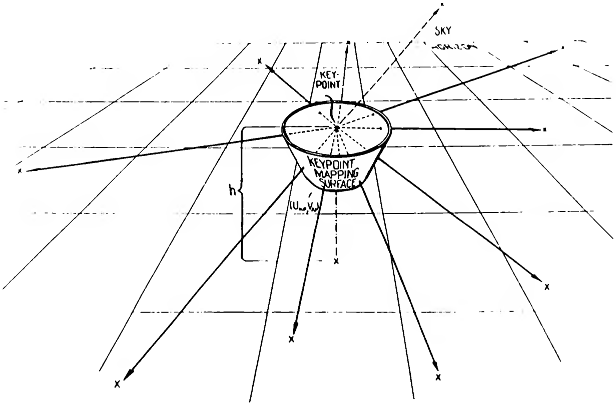


Figure 3: Ground-to-Keypoint Image Mapping

CREATION OF KEYPOINT IMAGES

Keypoint images are produced before simulation begins and stored on a high-capacity, rapid access storage device such as a videodisc. They are panoramic in the sense that they have a 360 x 90 degree field of view (see figure 3). Since the ground is absolutely flat, the keypoint image is conceptually a warped photograph (taken by an extremely wide field-of-view lens) of the entire ground plane out to the infinite horizon.

Mathematically, creation of a keypoint image is a mapping process. Rays are traced from the keypoint location through an imaginary keypoint surface until they intersect the ground, as in figure 4. Currently, IVEX corporation uses a polar projection surface (e.g., hemispherical, paraboloidal) for keypoint mapping in the VDS-1000 visual system. Specifically, the boundary of each keypoint pixel is traced on the ground thus forming a pattern of radial lines and concentric circles. The areas defined in this manner are the projections of the keypoint pixels onto the ground plane. Inside each area, the average values for each of the three color components (red, green, and blue) are computed and the resulting color vector is used as the value of that particular keypoint pixel.

This area sampling process is important in reducing aliasing artifacts since the digital keypoint image is a sampled version of the two-dimensional terrain. Because the polar keypoint surface sampling grid is non-uniform with respect to the rectilinear ground plane, we cannot avoid some aliasing in creating the keypoint image. Area sampling provides a means to convey some information to the keypoint image about every point on the terrain.

Keypoint images have better resolution of objects closer to the keypoint than those farther away. As the observer flies above the terrain, the visual system CPU requests the videodisc subsystem to load new keypoints into keypoint memory. The CPU attempts to select the "best" keypoint image to use at any given point in time in generating displayed images. There are many factors which are considered in keypoint selection but the basic trade-off is between blurry displayed images and increased aliasing. Image remapping as used by this technique can cause some areas of the displayed image to effectively magnify the keypoint image while other areas of the same displayed image may actually appear to locally reduce the keypoint image. This inconstant magnification implies

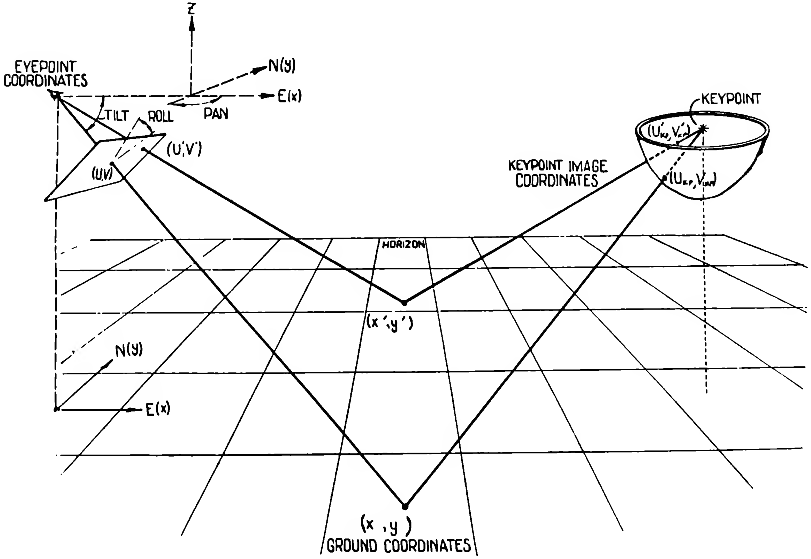


Figure 4: Display Screen-to-Keypoint Image Mapping

blurriness in the former case and aliasing in the latter. A third effect in practice is introduced by the search and retrieve times of the videodisc player. Since sometimes it may take several seconds to search for and load a keypoint image, the CPU must anticipate which keypoint will be needed next so that it will be available at the appropriate time.

Keypoint locations are arranged in layers of planar grids. As demonstrated in figure 5, the lateral spacing between keypoints on a particular plane is proportional to their altitude. Typically, the keypoint grids will look like an inverted pyramid over the runway, rising up to the minimum cruising altitude. The slope of the pyramid's sides will be the angle of the lowest allowable approach path. Since the observer will be permitted to fly anywhere above the minimum cruising altitude, grids of keypoints above this altitude must span the entire database. The spacing between keypoint grids can vary geometrically with keypoint altitude. Thus, the keypoint storage requirements are influenced most strongly by the size of the lower altitude keypoint grids.

IMAGE EXTRAPOLATION TECHNIQUE

A keypoint image is a representation of the entire world below the horizon. Therefore, it is possible to create new images derived from the keypoint image which represent the view from locations other than the keypoint. This method is called image extrapolation and is the fundamental principle behind the operation of the VDS-1000. Attention must be paid to proper sampling techniques in such a remapping procedure to avoid aliasing while maximizing scene information content.

Conceptually, the remapping can be considered to be performed in two steps. First, the mathematical transformation from points on the display screen to points on the ground is obtained. The ground and screen are both planar surfaces, so that the desired transformation is affine (due to perspective). This can also be viewed as a linear transformation in homogeneous coordinates. Let (U,V) be the coordinates of a pixel on the display screen where (U,V) = (0,0) is the center of the screen. Also let (X,Y) be a location on the ground where (X,Y) = (0,0) is the point on the ground directly under the keypoint. We can then write the transformation in the form

$$\begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} T11 & T12 & T13 \\ T21 & T22 & T23 \\ T31 & T32 & T33 \end{bmatrix} * \begin{bmatrix} U \\ V \\ 1 \end{bmatrix}$$

where the vector (A,B,C) is a representation of (X,Y) in homogeneous coordinates. In particular, X = A/C and Y = B/C. The elements of the matrix are functions of the observer pan, tilt, and roll angles and of the displacement of the eyepoint from the keypoint. The matrix elements are updated at frame rate.

This matrix operation can actually be efficiently implemented using accumulators instead of multipliers. Notice that as the displayed image is raster scanned, the U variable is incrementing by 1 every pixel while the V variable is incremented only at the beginning of each scan line. IVEX has developed a custom integrated circuit, the RXC-01, which can perform this accumulation and convert the result to floating point, if desired, for further processing. Thus, the entire matrix calculation shown above can be implemented using three IC's.

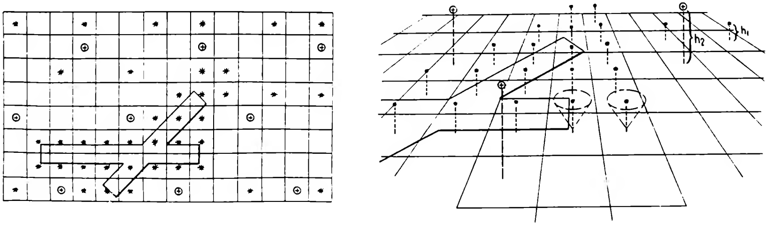


Figure 5: Typical Keypoint Placement

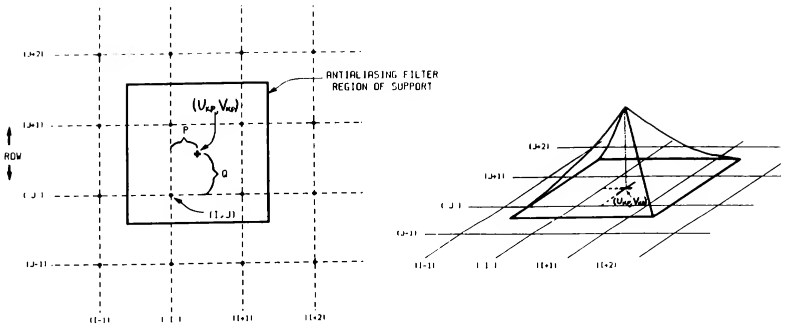


Figure 6: Antialiasing Filter

The second step of the remapping procedure involves establishing the transformation between the ground plane and the keypoint image. This is merely the inverse of the transformation used to originally generate the keypoint images. Let (UKP,VKP) be the coordinates in the keypoint image address space. It is important to note that UKP and VKP are functions of only A, B, and C. Thus, the mapping between the ground plane and the keypoint image can be implemented in special purpose hardware which does not require any frame rate parameters.

The above technique establishes a mapping from the displayed image to the keypoint memory address space. Effectively, we are using two functions UKP(U,V) and VKP(U,V) whose definitions can change at frame rate. The result is a constant-time algorithm in which the time to perform the procedure is unaffected by the contents of the database or the motion or location of the observer. Since the displayed image is created in a raster-scan fashion, pixels can be displayed as they are computed thereby eliminating the need for a frame buffer.

The computed values UKP and VKP are real numbers. In practice, they will have fractional components indicating that the ideal keypoint address falls between four surrounding keypoint image pixels. We can use the integer portions of UKP and VKP to identify these four pixels and the fractional components to determine how to weight the RGB contributions of these pixels to form the resulting pixel for display. Figure 6 shows a typical case of keypoint image addressing using real-valued addresses and sketches the impulse response for a simple 2 x 2 pixel antialiasing filter which works very well in this technique. This filter is a separable two-dimensional triangle filter and is the product of two one-dimensional triangle filters. It is worthwhile to note that such a filter can now be implemented with a single IC, such as the IVEX PWA-01, per color component.

ATMOSPHERIC EFFECTS

It is a fairly straightforward task to add range-based haze to images created by the above technique. With absolutely flat terrain, the problem of determining the distance from the eyepoint to the ground pixel-by-pixel can be formulated as an elementary geometry problem involving the intersections of lines with a plane. For each pixel, a ray is constructed from the eyepoint to the ground plane which travels through the given pixel. This is very similar to the first step of the image extrapolation process described above. The resulting distance is divided by the visibility distance specified by the user to obtain the visibility index. This index is then used (with an exponential extinction function) to determine how much haze to blend with the terrain pixel to obtain the hazed pixel.

In the VDS-1000, the user can specify independent haze distances for above and below horizon pixels. This permits simulation of a cloud layer through clever manipulation of these haze distance parameters. The observer can take off seeing completely overcast skies, watch the terrain haze out as he approaches the cloud bottom, and then see blue sky appear as he punches through the cloud top. Coordination of these effects is performed by the visual system CPU. Thus, a relatively simple software atmospheric model can greatly increase the flexibility of the total simulator. A number of effects can be modeled including broken clouds and ground fog.

SYSTEM ENHANCEMENTS

In the above sections, we have only discussed the use of image extrapolation to render the terrain. The technique will work equally well in generating a textured cloud layer. Although the actual textured surface must be absolutely flat, the haze generator can be used to gradually obscure the display as the cloud plane is approached. This adds the illusion of three-dimensionality to the cloud layer. If the visual system is configured with only a single address generator, then image extrapolation cannot be used on portions of the scene above and below the horizon at the same time. However, if independent image extrapolation systems are dedicated to above and below horizon image generation, then it is possible to texture the entire displayed image. In fact, the system could be adapted so that the cloud plane can be partially transparent thus allowing a high-altitude observer to see the terrain through the clouds.

The problem of simulating landing lights produced by the observer's own aircraft is actually quite similar to creating range-based haze. The pixel-by-pixel distance from the eyepoint to the ground computed by the haze generator is used to attenuate the beam power with increasing distance. If the landing lights can be approximated as emanating from the eyepoint location, then the apparent beam pattern will not change as the observer moves either linearly or angularly. Only the pixel-by-pixel intensity will change as a function of distance.

In general, combining the output of other graphics subsystems with the flat terrain is fairly straightforward. The geometry is much simpler than with three-dimensional terrain but, more importantly, any object with positive altitude will always occult the ground. This means that graphics from other sources, such as lightpoint and polygon generators, can be simply overlaid on top of the terrain scene. IVEX currently offers both of these devices in its flight simulator visual system product line. The lightpoint generator is a single board capable of producing 2000 raster lightpoints per frame time. Of course, the modeled light sources do not have to be confined to the ground plane--they can be placed arbitrarily in three dimensions. This device is programmable and supported light types include omnidirectional, half-plane directional, VASI, moving lights, flashing or pulsating lights, and stars. The polygon generator is also a single board device which can render several hundred three-dimensional polygons at frame rate. It is especially useful in simulating buildings and other aircraft but it also has a graphics mode which we have used to produce heads-up display (HUD) symbology.

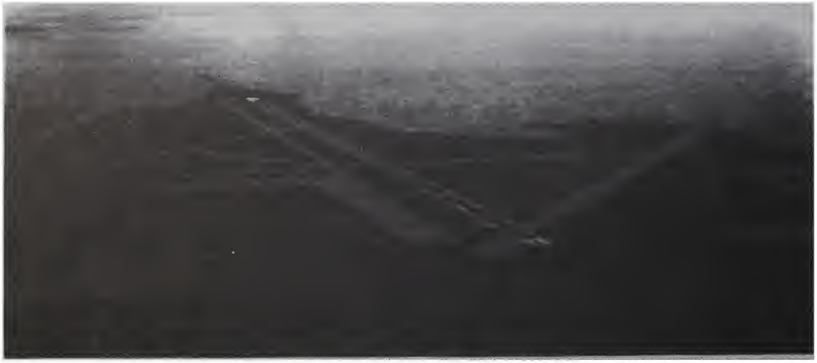


Figure 7: St. Petersburg Airfield

DYNAMIC TEXTURE IN VISUAL SYSTEM

88-4578-CP

Masaru Fujino
and
Masato Ogata

Mitsubishi Precision Co., Ltd. ,Kamakura, Japan

Abstract

In the current CGI visual system, texture mapping is used to enhance the detail of image. The state of the ocean waves and stirring of grass on the ground surface, however, are not sufficiently simulated. To improve the image, we have developed a system which computes linear combinations of some basic patterns, and using the computation results, a multi-stage color mixing is performed. Parameters specifying texture patterns are controlled to generate various dynamic textures. The image of dynamic ocean surface will produce effective training of ASW mission by helicopter.

Nomenclature

(a_i, b_i)	translation for basic pattern i, $1 \leq i \leq 8$ basic patterns are stored in 2D-array memory
C	color which consists of 3 components, R, G and B
i	the number of basic pattern
(j, k)	the address for 2D array of basic pattern memory
m	mapping from screen coordinate to polygon coordinate
(p, q)	the point to be textured on screen
s	weighted sum of texture value
t	texture value or element of basic pattern
(x, y)	the point to be textured on polygons
λ_i	wave length of basic pattern i

Introduction

It is considered that the Computer Image Generation, the main current of technology of present visual system will, because of its flexibility and versatility, continue to be the main current of technology for the 1990's. Recent CGI system manages the database including more than 100,000 faces or several 100,000 edges and is capable of realtime generation of 10,000 faces or several 10,000 edges. Rapid progress of the semi-conductor devices and development of new algorithms will enable the CGI to have the perfomance more than 10 times of present system. The amount of information of the real world to be simulated, even though the gaming area is limited to several hundred nautical miles square, is enormous: if the ground surface, the runway, the markings on the runway, the layer of the paint, heave of the ocean waves, the ripples, the white-crested waves, etc. are accurately modeled, 100,000 faces or 1,000,000 faces are too small to represent them. To fill the gap, utilization of texture technology is effctive. Texture represents the state and actuality of the ocean, terrain or the surface of the runway.

For fixed-wing aircraft simulation, traditional texture gives sufficient cue of actual height and actual speed to the pilots. For helicopter simulation, however, sufficient cue of height and speed have not been given. For training of low-altitude hovering or NOE flight by helicopter, detail expression of ocean surface or terrain is required: the ocean must have dynamic heave of waves, ripples and white-crested waves, and the effect of rotor washdown must be superimposed on the waves.

Texture Generation

We have developed a texture generator with an aim as described below:

(1) To represent dynamic ocean surface and terrain realistically.

(2) To make the hardware small and compact.

Fig.1 shows the architecture of our visual system including the texture generator.

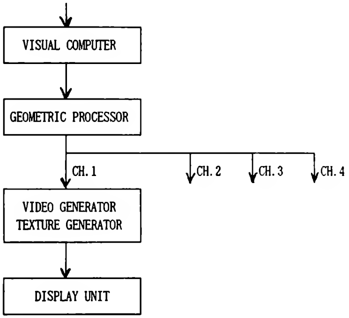


Fig.1 Visual system block diagram

Fig.2 shows the concept of the texture generator developed by us.
The process of the texture generator is as follows.

Transformation

$$(x, y) = m^*(p, q) \tag{1}$$

$$(j_i, k_i) = \lambda_i^*(x_i, y_i) \,|\, \mathrm{MOD}\ \lambda_i^* + (a_i, b_i)^* \tag{2}$$

* denotes that the parameter(s) is(are) controllable in realtime.

Computation of Linear Combinations

$$t_i = t_i\ (j_i,\ k_i) \tag{3}$$

$$S_A = \sum_i a_i \cdot t_i \tag{4}$$

$$S_B = \sum_i a_i \cdot t_i \tag{5}$$

$$S_C = \sum_i a_i \cdot t_i \tag{6}$$

Multi-stage Color Mixing

$$C = C_C \cdot S_C + C_D \cdot (1 - S_C) \tag{7}$$

$$C = C_B \cdot S_B + C\ \ (1 - S_B) \tag{8}$$

$$C = C_A \cdot S_A + C\ \ (1 - S_A) \tag{9}$$

As shown in Fig. 2 and Eq.s (1) - (9), linear combinations of basic patterns with independent scalings and translations are computed, and using the computation results, 3-stage color mixing is performed.
The 3-stage color mixing enables the generation of full color texture and various texture using the pattern memory, which contains limited number of the basic patterns. Many parameters are controllable, frame to frame or field to field. This architecture enables the generation of various dynamic texture: rustling of grass in the wind, dynamic wave of the ocean , explosion of missiles and motion blur.

Results and Conclusions

Fig. 3-8 show samples of image generated in realtime. Basic patterns in these scenes have been created by computation, or by photograph digitization or by hand-written picture digitization. The images of dynamic ocean surface showing the heave of the waves, the ripples, the white-crested waves and the water rings produced by rotor washdown of the helicopter are especially realistic, and produce effective training of ASW mission by helicopter. The method of texture synthesis was effective. The method also enables full color translucency by applying the background color as C_A, C_B, C_C or C_D, pixel to pixel.

Acknowledgement

The authors would like to acknowledge the helpful comments by Y. Sakurai, JASDF.

References

[1] W. K. Pratt, "Digital Image Processing," John Wiley & Sons, 1978.

[2] B. Schachter, "Long crested wave models", Computer Graphics and Image Processing, No. 12, pp. 187 - 201, 1980.

[3] A. Fournier and W.T. Reeves, "A Simple model of waves," Computer Graphics, vol. 20, No. 4, Aug. 1986 (SIGGRAPH '86 Conference Proceedings), PP 75 - 84.

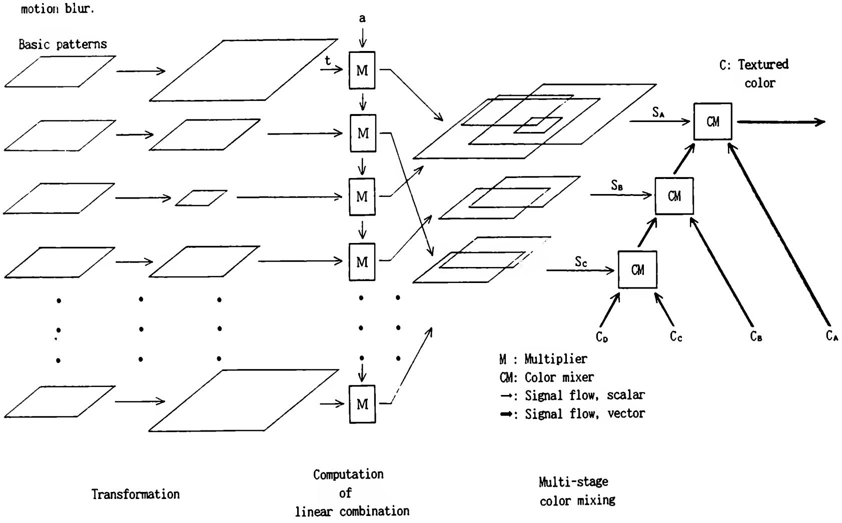


Fig. 2 Concept of the texture generator

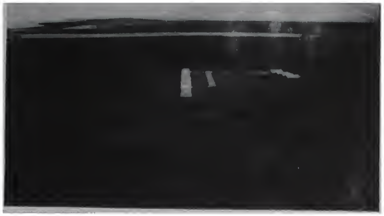


Fig. 3 Approaching to an airport.

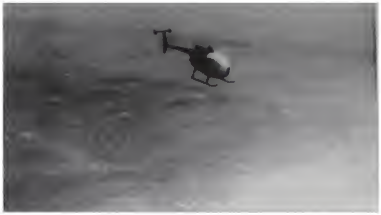


Fig. 6 The ocean surface showing the heave of the waves, the ripples, white-crested waves and water rings produced by helicopter rotor washdown.

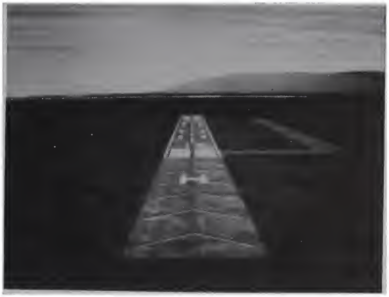


Fig. 4 Landing onto the runway.

Fig. 7 Approaching to a war-ship

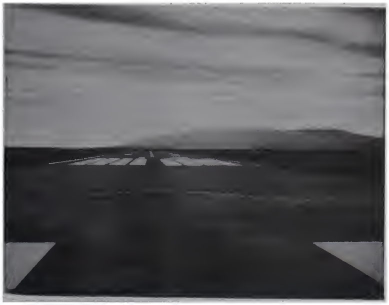


Fig. 5 On the runway

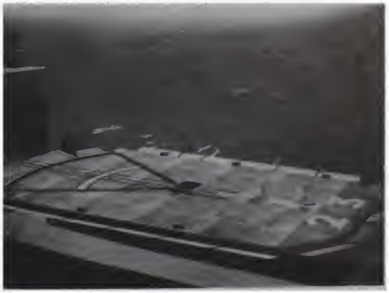


Fig. 8 Landing onto the war-ship.

MULTIPLE FRAME RATE INTEGRATION

88-4579-CP

A. Haraldsdottir
Applied Dynamics International, Ann Arbor, Michigan

and

R. M. Howe*
The University of Michigan, Ann Arbor, Michigan

Abstract

Dynamic systems can often be separated into fast and slow subsystems. The speed and accuracy of a simulation of such systems can frequently be improved by using a frame rate for numerical integration of the fast system which is an integer multiple of the frame rate used for the slow system. The technique of multiple frame-rate integration can be especially important in real-time simulation. In this paper the multiple frame-rate method is introduced, including techniques for converting slow data sequence outputs from slow subsystems to fast data sequence inputs for fast systems. The suitability of various integration algorithms for multiple framing is discussed. The implementation of multiple frame-rate integration using the simulation language ADSIM for the AD 100 computer is described, including software which allows, without program recompiling, choice of multiple-frame ratios and choice of different interpolation or extrapolation algorithms for slow-to-fast system interfacing. The paper concludes with an example of multiple framing applied to the simulation of a combined air frame and flight control system in order to improve both the accuracy and stability of the simulation.

1.Introduction

Dynamic systems can often be separated into fast and slow subsystems. One example is a combined air frame and flight control system, where the rigid airframe represents a slow subsystem, and both elastic structural modes and the flight-control system, including control-surface actuators, represent fast subsystems. Another example is a helicopter when modeled by the blade element method, where the rigid airframe again is the slow subsystem and the rotors are fast subsystems.

Multiple frame rate integration refers to the technique of making an integer multiple of integration passes through one or more fast subsystems for each pass through the slow subsystem. This reduces the integration step size for the fast subsystem. Since the dynamic errors in a digital simulation will be dominated by the integration truncation errors associated with the fast subsystem, the use of multiple framing can improve significantly the simulation accuracy for a given real-time processor. The accuracy improvement when using multiframing is much more substantial when the fast subsystem is considerably less complex and therefore requires much less processor time than the slow subsystem.

The overall concept of multiple frame rate integration is described in Section 2, along with the requirement to use extrapolation or interpolation to interface slow subsystems to fast subsystems. The section also introduces dynamic error measures, following which the compatibility of specific integration methods with multiple framing is discussed. Section 3 presents various interpolation and extrapolation algorithms for slow to fast data sequence conversion, as well as the dynamic errors associated with this conversion process.

Often it may not be clear exactly how a dynamic system should be partitioned into fast and slow subsystems in order to make most effective use of multiple framing. It may also be difficult to predetermine the optimal frame rate multiple for dynamic accuracy improvement. Analytic methods based on both time and frequency domain considerations, as introduced in Section 2, help in making these choices. However, in Section 4 an interactive software system is described which permits the user to experiment with different problem partitioning, frame rates, and interface extrapolation and interpolation methods. In Section 5 a combined air frame and flight-control system is used to illustrate the multiple framing analysis and synthesis techniques described in the earlier sections. Section 6 contains the concluding remarks.

2. Description of Multiple Frame Rate Integration

The separation of a dynamic system into slow and fast subsystems is illustrated in Figure 1. The slow system utilizes an integration step size denoted by T, whereas the fast system employs a step size denoted by h, where $h = T/N$ and N is an integer. Hereafter we will refer to N as the frame ratio. In Figure 1 the output data sequence $\{r_n\}$ with sample period T from the slow subsystem is converted to a fast data sequence $\{f_k\}$ with sample period h by means of an interpolator (or extrapolator). This is necessary to provide the fast subsystem with inputs having a sample period h equal to the fast subsystem integration step size. Examples of the generation of a fast sequence from a slow sequence are shown in Figure 2 for a frame ratio $N = 4$. In Figure 2a first-order interpolation is illustrated; in Figure 2b first-order extrapolation is used. Clearly the interpolation gives a more accurate result than extrapolation. In Section 3 we will see how we can quantify the dynamic accuracy of these and other interpolation and extrapolation algorithms in terms of equivalent gain and phase shift for sinusoidal data sequences.

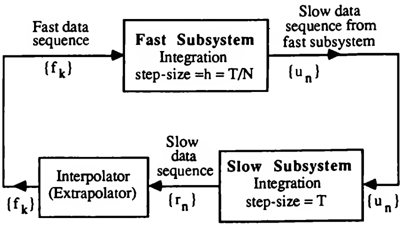
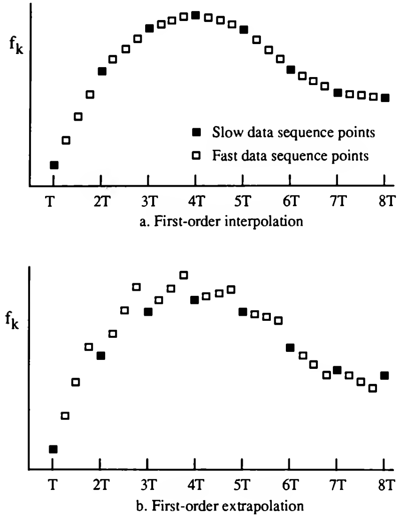


Figure 1. Multiple framing applied to a dynamic system.

*Professor, Department of Aerospace Engineering
Member AIAA

The output data sequence with sample period h generated by the fast subsystem in Figure 1 will need to be converted to a slow data sequence $\{u_n\}$ with sample period T in order to provide inputs for the slow subsystem integration algorithm. This conversion is easily accomplished by utilizing every Nth member of the fast data sequence output, although some multiple-pass integration methods may also require intermediate data points over the sample-period T.

From Figure 2 it is evident that interpolation of the data sequence $\{r_n\}$ over the nth sample period requires both r_n and r_{n+1}. Clearly r_{n+1} will only be available if the nth integration step in the slow subsystem has already been executed. On the other hand, extrapolation of the data sequence $\{r_n\}$ over the nth sample period requires r_n and r_{n-1}, both of which are available without prior execution of the nth integration frame in the slow subsystem. In this case, therefore, there is the option of performing the N fast subsystem integrations first with step size h, followed by the slow subsystem integration with step size T.



a. First-order interpolation

b. First-order extrapolation

Figure 2. Generation of a fast data sequence from a slow data sequence. $N = 4$.

Before discussing the suitability of various numerical integration algorithms for multiple framing, we review briefly the formulas and attributes of several of the candidate integration methods for real-time simulation. We will also describe dynamic error measures and stability considerations which will be important in understanding the benefits to be derived from multiple framing.

One of the most commonly used algorithms for real-time integration is the second-order Adams-Bashforth predictor method. The AB-2 formula for integrating the state equation $dx/dt = f[x,u(t)]$ is given by

$$x_{n+1} = x_n + h\left(\tfrac{3}{2}f_n - \tfrac{1}{2}f_{n-1}\right) \qquad (1)$$

where

$$f_n = f(x_n, u_n) \qquad (2)$$

Here $x_n = x(nh)$ and $u_n = u(nh)$, where h is the integration step size, n is an integer, x is the state variable, and $u(t)$ is the input, which is considered to be an explicit function of time t. The AB-2 formula is derived from the area under a linear extrapolation of the derivative f from $t = nh$ to $t = (n+1)h$ based on f_n and f_{n-1}. It is a second-order integration method, with dynamic errors proportional to h^2. When AB-2 integration is used to solve linear differential equations, it can be shown from z transform theory that the fractional error, e_λ, in any characteristic root λ is given approximately by the formula(1)

$$e_\lambda = \frac{\lambda^* - \lambda}{\lambda} \cong -\frac{5}{12}(\lambda h)^2, \quad |\lambda h| << 1 \qquad (3)$$

Here λ^* is the equivalent characteristic root of the digital system. To estimate the dynamic errors in using AB-2 integration for nonlinear differential equations, we can linearize the equation with respect to any reference or steady-state solution and then apply Eq. (3). For a given step size h the largest error will result from the characteristic root λ of largest magnitude. Thus Eq. (3) can be used to estimate the characteristic root errors in any simulation with known roots or eigenvalues λ. Or, if we have a given accuracy requirement on the roots, Eq. (3) permits us to establish the maximum allowable step size h when using AB-2 integration for that simulation.

Since Eq. (3) applies equally well for complex roots , it can also be used in this case to derive the following approximate formulas for the fractional error in root frequency, e_ω, and the damping ratio error, e_ζ(1):

$$e_\omega = \frac{\omega^* - \omega}{\omega} \cong \frac{5}{12}(1 - 4\zeta^2)(\omega_n h)^2, \quad \omega_n h << 1 \qquad (4)$$

$$e_\zeta = \zeta^* - \zeta \cong \frac{5}{6}(\zeta - \zeta^3)(\omega_n h)^2, \quad \omega_n h << 1 \qquad (5)$$

Here ω, ζ and ω_n represent the frequency, damping ratio and undamped natural frequency, respectively, associated with the complex root λ, while ω^* and ζ^* are the frequency and damping ratio, respectively, associated with the equivalent root λ^* of the digital system. Again, for any complex root pair Eqs. (4) and (5) can be used to estimate the frequency and damping ratio errors. Alternatively, for required accuracy in root frequency and damping ratio, Eqs. (4) and (5) can be used to establish the maximum allowable step size h when using AB-2 integration for the simulation. Eqs. (3), (4) and (5) will be useful in establishing the optimal frame ratio N when using multiple framing.

AB-3 and AB-4 predictor algorithms, based on second and third-order extrapolation, respectively, of the state-variable derivatives, can also be used as real-time integration methods. They produce dynamic errors proportional to h^3 and h^4, and formulas equivalent to Eqs. (3), (4) and (5) for characteristic root errors can be derived(1).

In addition to characteristic root errors it is important to consider the stability of integration algorithms, especially when using predictor methods. This is because the extrapolations based on past state-variable derivatives introduce extraneous states and hence extraneous roots into the simulation. If the step size h becomes too large, these roots can cause instability, even though the principal roots are accurately represented. In Figure 3 the stability boundaries for the AB predictor methods are shown in the λh plane. For any combination of step size h and eigenvalue λ outside these boundaries the simulation will be unstable. This can also become a very important reason for considering the use of multiple framing, as we shall see later in the example in Section 5.

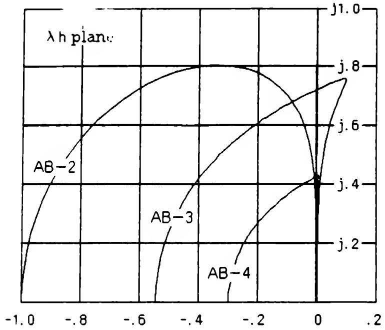


Figure 3. Stability boundaries for AB-2 integration.

In addition to the single-pass predictor-corrector algorithms considered above, real-time simulations can employ mutiple-pass integration algorithms such as Runge-Kutta methods. An example is the following RK-2 algorithm:

$$\hat{x}_{n+1/2} = x_n + \frac{h}{2} f(x_n, u_n) \qquad (6)$$

$$x_{n+1} = x_n + hf(\hat{x}_{n+1/2}, u_{n+1/2}) \qquad (7)$$

Euler (rectangular) integration is used for the first pass in Eq. (6) with a step size of $h/2$ to compute an estimate, $\hat{x}_{n+1/2}$, for the state halfway through the integration step. This estimate is then used in the second pass, along with the input $u_{n+1/2}$, to compute the derivative halfway through the step. In Eq. (7) this derivative is used to computex_{n+1}. Both the AB-2 algorithm in Eqs. (1) and (2), and the RK-2 algorithm in Eqs. (6) and (7), are compatible with real-time inputs, since in both cases the input u is not required for algorithm execution prior to its availability in real time. For this reason the above version of Runge-Kutta integration is often designated by RTRK-2. The more commonly used version of RK-2, frequently called Heun's method, employs Euler integration with a step size of h for the first step. The resulting estimate, $\hat{x}_{n+1}$, is then employed in the second pass, along with u_{n+1}, to compute x_{n+1} using trapezoidal integration. Since u_{n+1} is not yet available in real time at the start of the second pass, standard RK-2 is not compatible with real time inputs.

The RTRK-2 formula for e_λ, the fractional eror in characteristic root, is identical with that in Eq.(3) for AB-2 except that the coefficient 5/12 is replaced by the coefficient 1/6. For complex roots the RTRK-2 formulas for e_ω and e_ζ, the damping ratio error, are identical with those in Eqs. (4) and (5) for AB-2 except that the coefficients 5/12 and 5/6 are replaced by 1/6 and 1/3, respectively[1]. Thus the characteristic root errors associated with RTRK-2 integration for a given step size h are 40 percent of the AB-2 root errors. However, RTRK-2 is a two-pass method and will therefore take roughly twice as long to execute per integration step as AB-2. Since the dynamic errors vary as h^2, doubling the step size will quadrupal the errors. This more than offsets the error coefficient advantage of RTRK-2 over AB-2.

Another two-pass method is AM-2, the Adams-Moulton predictor-corrector algorithm. Here the first pass uses AB-2 integration to compute the predicted state, $\hat{x}_{n+1}$, which is employed along with u_{n+1} to calculate the derivative estimate $\hat{f}_{n+1}$. The second pass then uses trapezoidal integration to compute the final x_{n+1}. For AM-2 integration the approximate formulas for the charcateristic root errors are exactly -1/5th those shown in Eqs. (4), (5) and (6) for AB-2 integration. AM-2 is not compatible with real-time inputs because, as in standard RK-2, it requires u_{n+1} at the start of the second pass. However, a modified AM-2 method which computes $\hat{x}_{n+1/2}$ in the first pass can be used with real-time inputs[2].

There are two-pass predictor-corrector methods of higher order, e.g., AM-3 and AM-4, as well as three and four-pass Runge-Kutta methods (RK-3 and RK-4), any of which can be useful method for simulating dynamic systems. However, as noted below, the multiple-pass methods can complicate considerably the effective use of multiple framing.

From discussion thus far it is apparent that single-pass AB predictor algorithms are completely compatible with the multiple framing concepts described at the beginning of this section and illustrated in Figures 1 and 2. When we consider multiple-pass algorithms, however, the compatiblity is not so clear. For example, in the two-pass RTRK-2 method the first pass in Eq. (6) consists of a half-step Euler integration. When this is executed in the slow system, the second pass for the slow subsystem will require an input $u_{n+1/2}$ from the fast subsystem, as evident from Figure 1 and Eq. (7). How is this obtained with multiple framing from the fast subsystem? Does the fast subsystem execute N/2 two-pass RTRK-2 steps? If it does, the necessary fast subsystem inputs, if derived by interpolation from the slow subsystem, will need to be based on the rather inaccurate first Euler estimate $\hat{r}_{n+1/2}$. The other choice is to use extrapolation based on r_n and r_{n-1}, but this also may have accuracy problems. A question also arises regarding the second $N/2$ steps for the fast subsystem. Are these initiated from halfway through the slow frame, or from the beginning of the slow frame? Actually, it is probably more appropriate to use a single-pass method such as AB-2 for the fast subsystem, even though RTRK-2 is used for the slow subsystem. However, many choices and questions still arise. When a four-pass method such as RK-4 is used, the choices for the fast subsystem integration methods become even more numerous, although high order methods have been successfully employed with multiple framing[3]. Nevertheless, it is the opinion of the authors that single- pass methods are in general the algorithms best suited to multiple framing and lead to straightforward, easily understood mechanizations.

3. Extrapolation and Interpolation

In this section we consider interpolation and extrapolation algorithms as utililized in multiple framing to generate a fast data sequence from a slow data sequence. We first consider extraploation of a slow data sequence $\{r_n\}$ based on r_n and r_{n-1}. Let the extrapolation interval be denoted by the dimensionless parameter a. The corresponding extrapolation time interval is aT, where T is the sample period of the slow data sequence. The for r_{n+a}, the representation of $r(nT+aT)$ based on linear extrapolation, is given by

$$r_{n+a} = r_n + a(r_n - r_{n-1}) \tag{8}$$

In the example of linear extrapolation shown in Figure 2b, $N = 4$ and $a = 1/4$, $1/2$ and $3/4$ to generate the intermediate fast data sequence points from the current and past slow data-sequence points. To analyze the dynamic errors associated with Eq. (8) we consider the extrapolator transfer function for sinusoidal input data sequences and compare it with the ideal extrapolator transfer function. This is easily accomplished by using z transforms[4]. The z transform of Eq. (8) is

$$R_a^*(z) = [1 + a(1 - z^{-1})]R^*(z) \tag{9}$$

from which the extrapolator z transform, $H_e^*(z)$, is given by

$$H_e^*(z) = \frac{R_a^*(z)}{R^*(z)} = 1 + a(1 + z^{-1}) \tag{10}$$

The extrapolator transfer function for sinusoidal input data sequences is given by $H_e^*(e^{j\omega T})$. After dividing this by the ideal extrapolator transfer function, $H(j\omega) = e^{j\omega aT}$, we obtain the following formula for the fractional error in the extrapolator transfer function:

$$\frac{H_e^*(e^{j\omega T})}{H_e(j\omega)} - 1 = e^{-j\omega aT}[1 + a(1 - e^{-j\omega T})] - 1 \tag{11}$$

If the step size T is sufficiently small, the fractional transfer function error represented by the right side of Eq. (11) will be a complex number small in magnitude compared with unity. In this case it can be shown that the real part of the complex number is approximately equal to the fractional error in transfer function gain, and the imaginary part is approximately equal to the transfer function phase error[5]. When the exponential functions are expanded in power series with higher order terms neglected, the following approximate formulas are obtained for the first-order extrapolator transfer function gain and phase errors:

$$\begin{matrix}\text{Fractional}\\\text{gain error}\end{matrix} = e_M \cong \frac{a(1+a)}{2}(\omega T)^2 \ , \ \omega T << 1 \tag{12}$$

$$\begin{matrix}\text{Phase}\\\text{error}\end{matrix} = e_A \cong 0\cdot(\omega T)^2 \ , \ \omega\text{T} << 1 \tag{13}$$

Although Eqs. (12) and (13) are approximate, based on the dimensionless frequency ωT being small compared with unity, the formulas are surprisingly accurate up to $\omega T = 0.5$ for extrapolation intervals a over the range $0 < a < 1$. For the first order extrapolator it is clear that gain errors predominate over phase errors and are proportional to T^2.

Next we consider first-order interpolation based on r_n and r_{n+1}. Here the formula for r_{n+a} is given by

$$r_{n+a} = r_n + a(r_{n+1} - r_n) \tag{14}$$

Following the same procedure used for first-order extrapolation, we obtain the following formulas for the interpolator transfer function gain error:

$$\begin{matrix}\text{Fractional}\\\text{gain error}\end{matrix} = e_M \cong \frac{\text{a}(\text{a}-1)}{2}(\omega T)^2 \ , \ \omega T << 1 \tag{15}$$

As in the case of extrapolation, the interpolation transfer function phase errors are zero to order T^2, i.e., the gain errors predominate. Over the range $0<a<1$, the interpolator gain error of Eq. (15) is significantly less than the extrapolator gain error of Eq. (12). This is also apparent in comparing Figures 2a and 2b.

Second-order interpolation and extrapolation formulas can also be derived and analyzed in terms of transfer function errors using the above methodology[5]. When this is done, it is found that the approximate gain errors are zero to order T^3; the phase errors are proportional to $(\omega T)^3$ and predominate.

The overall dynamic errors generated by the fast data sequence using interpolation or extrapolation can be analyzed in the frequency domain by considering a slow data sequence $\{r_n\}$ that is sinusoidal with frequency ω. This analysis shows that the fast data sequence consists of a component at the input frequency ω, as well as components at $2(N-1)$ additional frequencies. If ω_0 is the sample frequency for the slow data sequence ($\omega_0 = 2\pi/T$), these additional frequencies will occur at $\omega_0 \pm \omega$, $2\omega_0 \pm \omega$, ... , $(N-1)\omega_0 \pm \omega$. Fortunately the influence of the additional freqencies will in general be small. This is because their amplitude relative to the input sinusoid will be of the same order as the fractional gain or phase error of the interpolator (or extrapolator) transfer function[5]. Thus the principal component in the fast data sequence will be at the input frequency ω. Compared with the ideal interpolator (or extrapolator) output, this component will represent a gain or phase error which is the simple average of the N gain or phase errors associated with the transfer functions for the N interpolation (or extrapolation) intervals a.

In Table 1 the gain or phase error coefficients for slow-to-fast data sequence conversion when using first and second-order interpolation, and zero, first and second-order extrapolation are tabulated for frame ratios $N = 2,3,4,5$, and ∞. Note that zero-order extrapolation is equivalent to no extrapolation, i.e., the fast data sequence consists of the slow data sequence value r_n repeated N times, then r_{n+1} repeated N times, etc. In this case the phase error predominates and is proportional to ωT. For large N the phase error for zero-order extrapolation approaches $-\omega T/2$. This is equivalent to inserting a time delay between the slow and fast subsystems equal to one half of the slow subsystem integration step size T. It is apparent that failure to use extrapolation or interpolation in multiple framing has the potential of introducing substantial dynamic errors.

4. Implementation in the Simulation Language ADSIM

The simulation language ADSIM is a high level software system designed for the AD 100 simulation computer. The form of the ADSIM language is very similar to other continuous system simulation languages such as ACSL, CSSL, and CSMP.

Table 1. Multiple Frame Interpolator and Extrapolator Gain and Phase Errors

Note: N = frame ratio. Gain and phase errors based on $\omega T << 1$

Phase error coefficient, $e_A/(\omega T)$

Inputs	$N = 2$	$N = 3$	$N = 4$	$N = 5$	$N = \infty$
r_n (zero-order)	-.2500	-.3333	-.3750	-.4000	-.5000

Gain error coefficient, $e_M/(\omega T)^2$

	$N = 2$	$N = 3$	$N = 4$	$N = 5$	$N = \infty$
r_n, r_{n-1}	.1875	.2593	.2969	.3200	.4167
r_{n+1}, r_n	-.0625	-.0741	-.0781	-.0800	-.0833

Phase error coefficient, $e_A/(\omega T)^3$

	$N = 2$	$N = 3$	$N = 4$	$N = 5$	$N = \infty$
r_n, r_{n-1}, r_{n-2}	.1563	.2222	.2578	.2800	.3750
r_{n+1}, r_n, r_{n-1}	-.0313	-.0370	-.0391	-.0400	-.0417
$r_{n+1}, r_n, \dot{r}_n$	-.0104	-.0123	-.0130	-.0133	-.0139

Since ADSIM and the AD 100 have been designed to be especially effective for high-speed real-time simulation, integration with ADSIM is performed using fixed-step algorithms. The standard integration methods provided in the language are Euler, Adams-Bashforth 2,3 and 4, two-pass Adams-Moulton 2,3 and 4, Runge-Kutta 2 and 4, and real-time Runge-Kutte 2 and 3, as introduced in Eqs. (6) and (7). The default integration method used in ADSIM is AB-2. At run time the user can, without recompiling, interactively select any of the other integration methods for any or all of the state variables. He/she can also program his own integration algorithms as difference equations. In ADSIM the user writes differential equations in first-order state-variable form. The equations are solved in a segment of code termed a DYNAMIC BLOCK. The user code in the DYNAMIC BLOCK is invoked by a routine called SIMEXEC, which executes on the AD 100 computer and performs execution control and integration.

To implement multiple frame rate integration on the AD 100 and make it as easy to use as possible, the routine SIMEXEC has been modified to handle the additional execution control needed. Since initialization and integration in ADSIM is performed on an entire DYNAMIC BLOCK, the modified routine SIMEXEC expects the differential equations to be divided into two blocks, one corresponding to the slow subsystem and the other to the fast subsystem. During each simulation frame the right hand sides of all equations in the slow DYNAMIC BLOCK are evaluated and integrated once, while equations in the fast DYNAMIC BLOCK are evaluated and integrated N times, where N is the frame ratio. The user can interactively control N at run time. Figure 4 illustrates the execution flow of the modified SIMEXEC routine. The code segments "sync1" through "sync7", "initial" and "terminal" represent optional code to allow the user to perform special calculations before and after the simulation is performed. The routine $DER evaluates the state-variable derivatives in the dynamic equations, and $INT and $INC perform integration of continuous equations and updating of discrete-time (i.e., difference) equations.

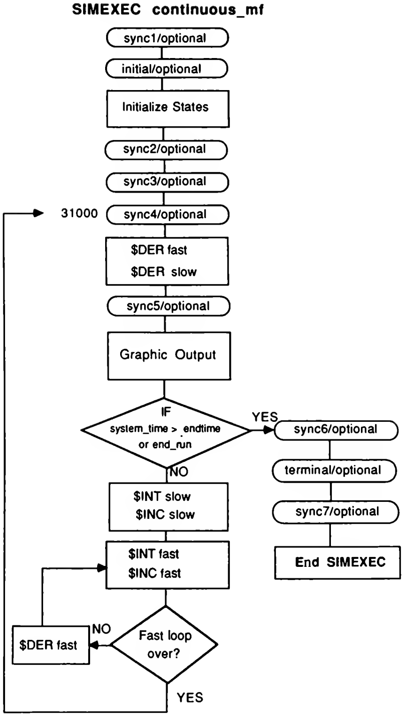


Figure 4. SIMEXEC for multiple frame rate integration.

In the dynamic block containing the fast subsystem the user needs to invoke a subprogram to perform interpolation on each of the slow variables used as inputs to the fast subsystem. Five subprograms for interpolation are supplied as standard functions with the ADSIM software package. Figure 5 illustrates the ADSIM code required to simulate a fourth-order system with multiframing.

To summarize, the user wishing to simulate a dynamic system using multiple frame rate integration needs only supply:

- The dynamic equations separated into slow and fast subsets.
- A subprogram invocation for each slow state variable to be interpolated.
- The number of fast integration steps for each slow step, i.e., the frame ratio N.

```
TITLE Two time scale model with multiframing
!
REGION initial
      frame_ratio_inv = 1./frame_ratio
END REGION
!
DYNAMIC fast
x3_inter = INTN1N(frame_ratio_inv,     %
                    slow_sample,x3,initialize)
e     =  r - kf*x3_inter
x1'   =  x2
x2'   =  -2.*z1*w1*x2 - w1*w1*x1 + k1*e
END DYNAMIC
!
DYNAMIC slow
x3'   =  x4
x4'   =  -2.*z2*w2*x4 - w2*w2*x3 + k2*x1
END DYNAMIC
!
DATA   z1          =    .3     %
       w1          =    35.    %
       k1          =    1000.  %
       z2          =    .9     %
       w2          =    .3     %
       k2          =    1.     %
       kf          =    1.     %
       r           =    1.
!
EXECUTE continuous_mf
```

Figure 5. ADSIM code to simulate a fourth-order system using multiframing.

The interactive real-time simulation features of ADSIM are preserved in the multiframing code. The variable "frameratio" can be changed interactively, and the actual computer execution time for one simulation frame is automatically measured and displayed after each change. The integration step size *T* divided by the frame execution time represents the execution speed with respect to real time (i.e., speedup over real time). This ratio is also calculated automatically after each change in multiframing parameters. When the ratio exceeds unity, the user knows the problem can be run in real time.

5. Multiframing Example

The use of multiple frame rate integration will be illustrated here with a reasonably large model that contains a small subsystem with fast dynamics. The model represents a typical business jet aircraft and includes a large data base associated with a number of multivariable aerodynamic functions, as well as a simplified flight control system. The aircraft is modeled as a rigid body with six degrees of freedom, three translational and three rotational. For the purpose of our multiframing example, only the aircraft pitch control loop will be considered. Figure 6 shows a block diagram of this control loop, where the inner feedback loop with the state variables δ_{es} and $\dot{\delta}_{es}$ is associated with the elevator control surface actuator.

Even though the overall airframe/flight-control system equations are nonlinear, it is useful to know the eigenvalues of the linearized equations in order to apply the stability and characteristic root error analysis discussed in Section 2. The eigenvalue computation is accomplished as follows: With the overall simulation frozen at any time, each state variable is perturbed by a small increment. From the resulting change in the state variable time derivatives, the partial of all state derivatives with respect to all state variables is computed numerically. From the Jacobian thus obtained the eigenvalues of the system are calculated. For a test flight condition of Mach 0.72 at 40,000 feet altitude the following eigenvalues were obtained:

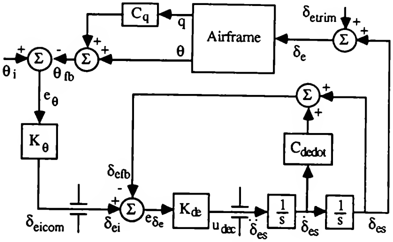


Figure 6. Pitch control loop.

$\lambda_{1,2} = -15.92 \pm j26.37$ $(\omega_n = 30.80, \zeta = 0.517)$

$\lambda_{3,4} = -1.274 \pm j4.674$ $(\omega_n = 4.844, \zeta = 0.263)$

$\lambda_5 = -1.000$

$\lambda_6 = -0.338$

$\lambda_7 = -0.016$

In the elevator actuator loop $C_{dedot} = 0.033$ and $K_{de} = 1000$. This makes the actuator natural frequency equal to 31.62 rad/s and the damping ratio 0.522. When the pitch-control loop is closed, this is the origin of $\lambda_{1,2}$ pair shown above. The other roots, which originate from the airframe poles and the quaternion stabilization loop, are substantially lower in magnitude. Clearly the pitch control loop is the fast subsystem which should benefit from multiple framing. Thus the partitioning of the overall model into fast and slow subsystems is straightforward in this example. In Figure 6 the airframe block represents the slow subsystem with the rest of the diagram representing the fast subsystem. The pitch loop equations are incorporated in the fast DYNAMIC BLOCK along with the subprogram invocations for the slow state variable interpolations, as shown in Figure 7. Here the state variables are *dedot* and *des*, and the fast subsystem inputs for interpolation are the pitch rate, *q*, and the four quaternions, *e1*, *e2*, *e3*, and *e4*. The quaternions are used to calculate the direction cosines *a13*, *a23*, and *a33*, from which the pitch angle *theta* (i.e., θ), is computed. If Euler angles rather than quaternions had been used in the simulation, then only *q* and *theta* would need to have been interpolated.

```
DYNAMIC fast
!
!       Elevator actuator:
!
!       Pitch control system:
!
deicom  =  Ktheta*(thetai-theta-Cq*q_int)
dei     =  Deilim*SAT(deicom*Ideilim)
e1_int  =  INTN1N(frame_ratio_inv,slow_sample,%
                  e1,initialize)
e2_int  =  INTN1N(frame_ratio_inv,slow_sample,%
                  e2,initialize)
e3_int  =  INTN1N(frame_ratio_inv,slow_sample,%
                  e3,initialize)
e4_int  =  INTN1N(frame_ratio_inv,slow_sample,%
                  e4,initialize)

e1e1    =  e1_int*e1_int
e2e2    =  e2_int*e2_int
e3e3    =  e3_int*e3_int
e4e4    =  e4_int*e4_int
a13     =  2.*(e2_int*e4_int-e1_int*e3_int)
a23     =  2.*(e2_int*e3_int+e1_int*e4_int)
a33     =  e1e1+e2e2-e3e3-e4e4
theta   =  ATAN2(-a13,SQRT(a23*a23+a33*a33))
q_int   =  INTN1N(frame_ratio_inv,slow_sample,%
                  q,initialize)
udec    =  Kde*(dei-des-Cdedot*dedot)
dedot'  =  Udelim*SAT(udec*Iudelim)
des'    =  dedot

END DYNAMIC
```

Figure 7. The fast DYNAMIC BLOCK in the multiframe simulation.

The benefit of multiple frame rate integration on both solution accuracy and numerical stability will be demonstrated here. Several interpolation algorithms were used for generating the fast data sequences, ranging from zero-order (no interpolation) to second order formulas. To study the accuracy improvement with multiframing, a reference solution was generated for pitch flight control response with an integration step size $T_1 = 0.000050$s. The pitch angle input θ_i was a step at $t = 0$ with a magnitude of 0.15 radian. AB-2 integration was used. Figure 8 shows the time history of the pitch angle response, θ, and the elevator actuator output, δ_{es}.

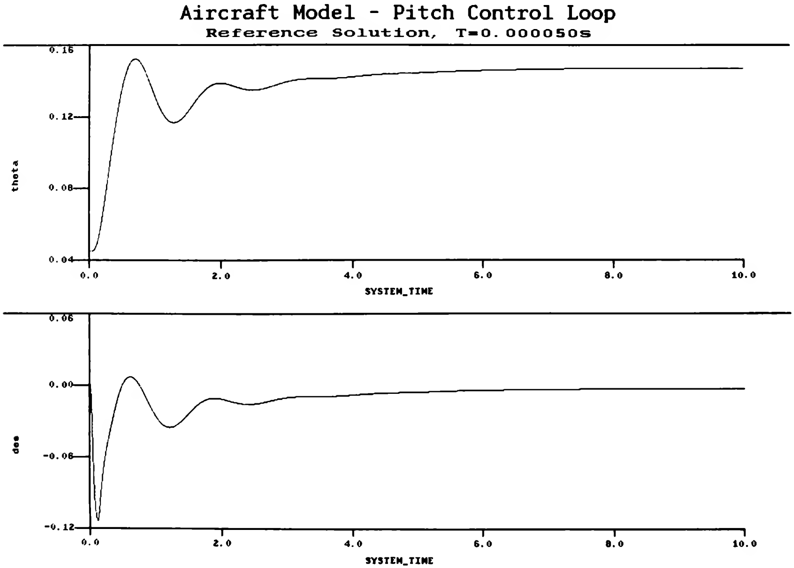


Figure 8. Reference step response solution for the pitch control system.

Time histories for a step size T_2 = 0.01s were then generated for frame ratios N ranging from 1 (no multiple framing) to 7. The error measure was computed as the mean absolute error in the elevator actuator output, δ_{es}. Table 2 shows the solution error for different frame ratios. It is clear from this data that multiple frame rate integration improves the solution accuracy substantially for frame ratios N up to 4. Beyond N = 4 there is little improvement.

Table 2. Solution Error for Different Frame Ratios with AB-2 Integration; Interpolation Based on r_{n+1} and r_n.

Frame Ratio	Mean absolute error in δ_{cs}
1 (no framing)	0.0004725 rad
2	0.0001461
3	0.0000952
4	0.0000821
5	0.0000794
6	0.0000792
7	0.0000791

The results in Table 2 can be understood by applying Eqs. (4) and (5) for the AB-2 characteristic root errors to the system eigenvalues listed earlier in this section. When this is done for $\lambda_{1,2}$ = -15.92 ± j 26.37 with a step size h = 0.01, the predicted fractional error in frequency e_ω = -0.00273 and the damping ratio error e_ζ = 0.0300. For $\lambda_{3,4}$ = -1.274 ± j4.674 and h = 0.01, e_ω = 0.00071 and e_ζ = 0.00048. Thus when no multiframing is used, the errors for eigenvalues $\lambda_{1,2}$ associated with the pitch control loop are much larger than the errors for the eigenvalues $\lambda_{3,4}$ associated with the airframe.

On the other hand when N = 5, the pitch control loop has a step size given by 0.01/5 = 0.002. Applying Eqs. (4) and (5) to $\lambda_{1,2}$ with h = 0.002 gives e_ω = -0.00011 and e_ζ = 0.0012. The larger of these, e_ζ, is now roughly comparable with the e_ω (= 0.00071) and e_ζ (= 0.00048) as obtained above for the roots $\lambda_{3,4}$ associated with the airframe system, which still utilizes the step size 0.01. For N greater than 5 the airframe roots will actually predominate and nothing is gained by multiframing the pitch control loop to reduce its step size.

To examine the effect of multiple frame rate integration on the numerical stability, the aircraft simulation was executed using AB-2 integration with five different interpolation algorithms and various frame ratios. In ADSIM the variables "frametime, "steptime" and "speedup" are used to show the relationship between integration step size and real-time execution. "Frametime" is the actual time in seconds required for the AD 100 to execute a single pass through the dynamic equations, i.e., one pass through the slow subsystem and N passes through the fast subsystem. "Steptime" is the mathematical step size used for the integration step T in the slow subsystem. "Speedup" is therefore defined as

speedup = steptime/frametime

and represents the speedup factor over real time in the AD 100 solution for the stepsize T being used.

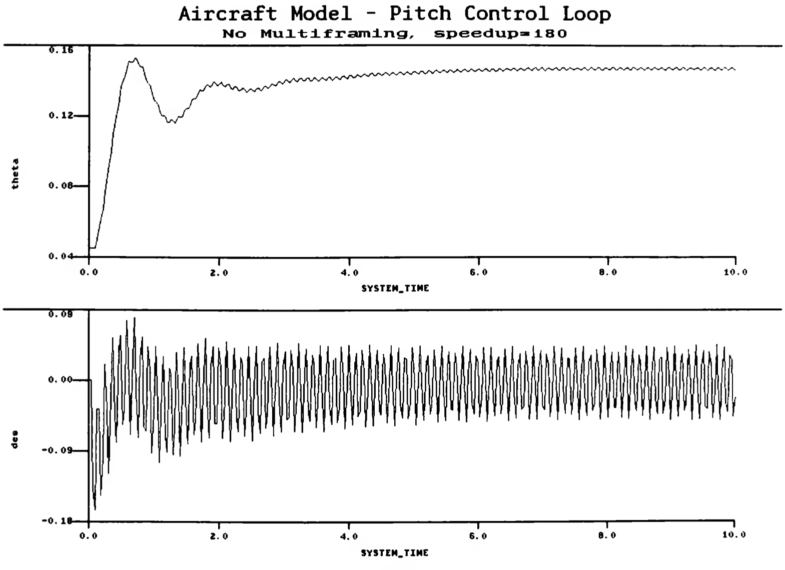


Figure 9. Marginally stable solution.

In the numerical stability experiment, marginal stability was determined by observing a graphical output of the pitch angle θ in response to the step input $\theta_i = 0.15$ radian. Figure 9 shows a typical time history for a marginally stable solution with the frame ratio $N = 1$, i.e., no multiple framing. Table 3 lists the integration step size T and corresponding speedup factor at which the solution becomes marginally stable for frame ratios ranging from 1 to 7 and for five different interpolation algorithms. Note that interpolation based on r_n is zero-order and hence represent no interpolation.

Comparison of the performance in Table 3 for various interpolation algorithms shows that first-order interpolation based on r_{n+1} and r_n gives the best performance. The improved dynamics of the higher-order interpolation algorithms is more than offset by their increased execution time. It is also clear that zero-order interpolation based on r_n, i.e., non interpolation, produces inferior results. This is because of the additional time delay, $T/2$, introduced between the slow and fast subsystems by zero-order interpolation, as noted in Section 3.

Table 3. Integration Step Size and Speedup Factor for Marginally Stable Solution.

Basis for interpolation	Integration step size T (speedup over real time)						
	Frame ratio $N = 1$	$N = 2$	$N = 3$	$N = 4$	$N = 5$	$N = 6$	$N = 7$
r_n	.02968 (187)	.05826 (306)	.08307 (374)	.1003 (395)	.1114 (390)	.1158 (365)	.1169 (335)
r_{n+1}, r_n	.02987 (183)	.05838 (287)	.08722 (361)	.1150 (411)	.1437 (452)	.1539 (432)	.1578 (400)
r_{n+1}, r_n, r_{n-1}	.02976 (169)	.05833 (259)	.08640 (315)	.1155 (357)	.1442 (387)	.1522 (361)	.1520 (323)
$r_{n+1}, r_n, \dot{r}_n$	.02978 (176)	.05814 (275)	.08673 (342)	.1154 (390)	.1437 (425)	.1521 (400)	.1521 (360)
$r_{n+1}, r_n, \dot{r}_n, r_{n-1}$	.02980 (168)	.05832 (256)	.08652 (311)	.1153 (351)	.1440 (380)	.1524 (335)	.1545 (322)

For $N = 1$ (no multiple framing) the table shows that the system becomes marginally stable for $T = h = 0.02968$. Here the instability is due to the eigenvalues $\lambda_{1,2} = -15.92 \pm j\,26.37$, as listed earlier in this section and associated with the dynamics of the actuator. Thus $\lambda_1 h = -0.473 + j\,0.783$, which indeed lies on the AB-2 stability boundary in Figure 3. This confirms that the instability with no multiframing originates in the fast subsystem. From z transform theory it turns out that the predicted frequency of instability in this case is equal to $0.270/T$ or 9.10 hertz, which is precisely the frequency of the undamped transient in Figure 9. On the other hand, for $N = 7$ and interpolation based on r_{n+1} and r_n, the system becomes marginally stable for $T = 0.1578$. In this case the step size of the fast subsystem is given by $h = T/7 = 0.0225$, which lies well within the AB-2 stability boundary for its roots $\lambda_{1,2}$. Thus the instability must result from the roots $\lambda_{3,4} = -1.274 \pm j\,4.674$ associated with the slow subsystem. In this case $\lambda_3 T = -0.201 + j\,0.738$, which indeed lies just within the AB-2 stability boundary in Figure 3 and confirms the expected instability based on the slow subsystem eigenvalues.

When multiframing is used for the fast subsystem, the "frametime" will actually increase. This is because the time required for the input interpolations and N integration steps of the fast subsystem is added to the execution time for the single integration step of the slow subsystem. Even though the overall frame time has increased, however, the integration step h for the fast subsystem will be reduced as the frame ratio N is increased, since $h = T/N$. This accounts for the improved stability evident in Table 3 for frame ratios up to $N = 5$, as reflected by the increased speedup over real time before the simulation becomes marginally stable. Beyond $N = 5$ the fast subsystem step size h has been reduced to the point where it is the slow system step size T that causes the instability. This is why the overall stability deteriorates for N larger than 5, since as N is further increased, the overall frame time and hence T for the slow subsystem actually increases.

In the example of mutiple frame rate integration described in this section, we have only considered interpolation for interfacing the fast subsystem to the slow subsystem. Alternatively, extrapolation could have been used. However, reference to Table 2 shows that the dynamic performance of extrapolation is considerably poorer. It should probably be considered only when it is desirable to avoid the necessity of integrating the slow subsystem first. There would, however, have been one advantage in using extrapolation for our example here. If extrapolation had been used, it would only have been necessary to interface the pitch angle θ and pitch rate q from the slow to the fast subsystem, rather than the four quaternions e_1, e_2, e_3 and e_4, as well as q. This would this have saved not only three interpolation calculations but also the additional calculation of direction cosines and θ each fast frame. When the interface between slow and fast subsystem consists entirely of state variables, this advantage for extrapolation is of course no longer present.

6. Conclusions

In this paper we have seen how the use of multiple frame rate integration in the simulation of a dynamic system can improve both the accuracy and stability of the simulation. In many cases it can make the difference between whether or not a simulation on a given computer can be run in real time with satisfactory accuracy. Analytic methods based on linearization of the dynamic system being simulated can be used to estimate the dynamic performance and stability for given integration methods and step sizes. This can in turn be utilized to make preliminary assessments of the optimal separation into slow and fast subsystems, as well as the optimal ratio N of the fast compared with the slow subsystem. However, it is also important to have an interactive software system which permits the user to experiment with the simulation itself in order to determine optimal problem partitioning and frame ratios. The application of such a system, ADSIM for the AD 100 computer, to the simulation of a combined airframe, flight control system with the use of multiple framing has been described. An additional consideration of importance in mutiple framing is the type of interpolation or extrapolation algorithm used for the slow-to-fast system interfacing. Again, analytic techniqes have been described here for assessing the performance of various algorithms, although interactive experimentation with different interpolation or extrapolation methods at run time can also be extremely useful.

7. References

1. Howe, R.M., "Transfer Function and Characteristic Root Errors for Fixed-Step Integration Algotithms," *Trans. of the Society for Computer Simulation*, Vol.2, No. 4, Dec., 1985, pp 293-320.

2. Howe, R.M., "The Role of Modified Euler Integration in Real-time Simulation," *Proceedings of the Conference on Aerospace Simulation III*, San Diego, 1988; pp 277-285. The Society for Computer Simulation, P.O. Box 17900, San Diego, CA 92117.

3. Palusinski, O.A., "Simulation of Dynamic Systems Using Multirate Integration Techniques," *Trans of the Society for Computer Simulation*, Vol.2, No. 4, Dec., 1985, pp 293-320.

4. Gilbert, E.G., "Dynamic Error Analysis of Digital and Combined Digital-Analog Systems," *Simulation*, Vol. 6, No. 4, April, 1966, pp 241-257.

5. Howe, R.M., "Dynamics of Digital Extrapolation and Interpolation," *Trans. of the Society for Computer Simulation*, Vol.2, No. 1, Dec., 1985, pp 169-187.

88-4580-CP

SOME IMPROVED METHODS FOR REAL-TIME INTEGRATION OF STATE VARIABLE DERIVATIVES WITH DISCONTINUITIES

R. M. Howe* and A. Nwankpa**

The University of Michigan
Ann Arbor, Michigan

Abstract

In this paper we consider an analytic averaging technique for integration of discontinuous nonlinear functions. Functions with both displacement and slope discontinuities are treated. The analytic averaging method is shown to provide much better accuracy than conventional integration algorithms. This is especially true when fixed integration step sizes are used, as in real-time simulation. A simple but practical example of a bang-bang control systems is used to verify the superior performance of the analytic averaging method. It is also shown how averaging formulas for unit step and unit ramp nonlinear functions can by superposition be used to construct analytic averaging formulas for any nonlinear function which has displacement and slope discontinuities. A modified form of Euler integration is shown to be especially compatible with the analytic averaging method.

1.Introduction

In real-time simulation of dynamic systems the time derivatives of state variables sometimes have discontinuities. For example, this is clearly the case when simulating a spacecraft attitude control system which uses on-off reaction control thrusters. It is also true in the simulation of continuous controllers with effort limiting, controllers with dead-zone, etc. In general the discontinuities occur at times which are asynchronous with respect to integration step times. Because of this the use of conventional integration methods can result in substantial dynamic errors. Methods have been proposed using variable integration step size to improve the accuracy when discontinuities are present[1,2,3]. However, in real-time simulation the integration step size must be fixed and the errors introduced by discontinuous derivatives can become very serious unless the step size is made inordinately small. A technique compatible with real time simulation which utilizes an intermediate step to the discontinuity has been described and shown to exhibit high accuracy[4]. However, this method can require considerable computation time when many discontinuous functions are present in a simulation.

A less accurate but faster method for handling discontinuous nonlinear functions in fixed step integrations has also been described[5]. The method, which uses an analytic averaging technique, is introduced in the next section. A general formula for the analytic averaging function for any nonlinearity consisting of straight line segments and displacement discontinuities is developed in Section 3 from averaging formulas for unit step and unit ramp nonlinear functions. Section 4 reviews some interpolation and extrapolation methods which are useful in applying the analytic averaging method. Finally, in Section 5 we present a simple but practical example of a bang-bang control system with hysterisis to demonstrate the superior performance of the analytic averaging technique. Example solutions are shown for AB-2 integration as well as a modified form of Euler integration which turns out to be especially compatible with the analytic averaging method.

2.Derivation of the Analytic Averaging Function

Assume that a dynamic system contains a scalar state equation given by

$$\frac{dy}{dt} = f(x) \tag{1}$$

where $f(x)$ is a nonlinear function which can include discontinuities in displacement or slope. Let h be the numerical integration step. Then the ideal numerical integration formula is given by

$$y_{n+1} = y_n + \int_{nh}^{(n+1)h} f(x)dt = y_n + \int_{x_n}^{x_{n+1}} \frac{f(x)}{dx/dt}dx \tag{2}$$

Here y_n represents $y(nh)$ and x is a function of time. We next assume over the interval of integration that the time derivative of x, dx/dt, is a constant which can be approximated by the following central difference:

$$\frac{dx}{dt} = \frac{x_{n+1} - x_n}{h} = \text{constant} \tag{3}$$

Then Eq. (2) can be rewritten as

$$y_{n+1} = y_n + f_{ave}h \tag{4}$$

where

$$f_{ave} = \frac{1}{x_{n+1} - x_n}\int_{x_n}^{x_{n+1}} f(x)dx \tag{5}$$

With Eq. (5) we have converted the integral of $f(x)$ with respect to t in Eq. (2) to an integral of $f(x)$ with respect to x in Eq. (5). In fact f_{ave} represents simply the average value of $f(x)$ over the interval of integration. For any specified $f(x)$ the integral is a function of x_{n+1} and x_n. It can be precomputed analytically when $f(x)$ is an analytic function of x. For example, if $f(x) = x$, a linear function, the f_{ave} function is given by

$$f_{ave} = \frac{1}{x_{n+1}-x_n}\int_{x_n}^{x_{n+1}} x\,dx = \frac{x_{n+1}^2 - x_n^2}{2(x_{n+1}-x_n)} = \frac{x_{n+1}+x_n}{2} \tag{6}$$

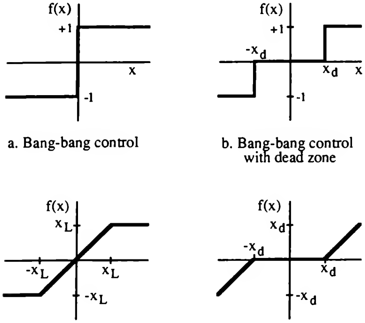
* Professor, Department of Aerospace Engineering
Member AIAA

** Department of Aerospace Engineering

In this case we see that Eq. (4) represents trapezoidal integration. However, when $f(x)$ is a nonlinear function of x, Eqs. (4) and (5) will produce a result which is more accurate than trapezoidal integration.

It should be noted that f_{ave} as given by Eq. (5) is undefined for $x_{n+1} = x_n$. In this case it is clear that f_{ave} should be equal to f_n, the value of $f(x)$ for $x = x_n$. If it is possible for x_{n+1} to be equal to x_n in a simulation, it may be necessary to add an "if" statement in the program in order to set $f_{ave} = f_n$ when $x_{n+1} - x_n = 0$.

Figure 1 illustrates some typical discontinuous nonlinear control functions. We now proceed to derive the f_{ave} formulas for these functions.



a. Bang-bang control

b. Bang-bang control with dead zone

c. Linear control with effort limiting

d. Linear control with dead zone

Figure 1. Typical controller nonlinearities.

Consider first the bang-bang switch function shown in Figure 1a. The function can be represented analytically by the formula $f(x) = x/|x|$. From Eq. (5) it follows that

$$f_{ave} = \frac{1}{x_{n+1}-x_n}\int_{x_n}^{x_{n+1}} \frac{x}{|x|}\,dx = |x|\Big]_{x_n}^{x_{n+1}} = \frac{|x_{n+1}| - |x_n|}{x_{n+1}-x_n} \qquad (7)$$

Note that if both x_{n+1} and x_n are positive, $f_{ave} = +1$. When both x_{n+1} and f_n are negative, $f_{ave} = -1$. When x_{n+1} and x_n have opposite polarity, f_{ave} will be somewhere between +1 and -1 and will represent the average value of the switch function over the interval x_n, x_{n+1}.

Next consider the linear function with limiting, as shown in Figure 1c. Here the function can be represented analytically by $f(x) = (|x+x_L| - |x-x_L|)/2$. From Eq. (5) we obtain the following formula for f_{ave}:

$$f_{ave} = \frac{1}{x_{n+1}-x_n}\int_{x_n}^{x_{n+1}} \frac{|x+x_L| - |x-x_L|}{2}\,dx$$

or

$$f_{ave} = \frac{1}{4(x_{n+1}-x_n)}\Big[(x_{n+1}+x_L)|x_{n+1}+x_L| - (x_n+x_L)|x_n+x_L|$$
$$- (x_{n+1}-x_L)|x_{n+1}-x_L| + (x_n-x_L)|x_n-x_L|\Big] \qquad (8)$$

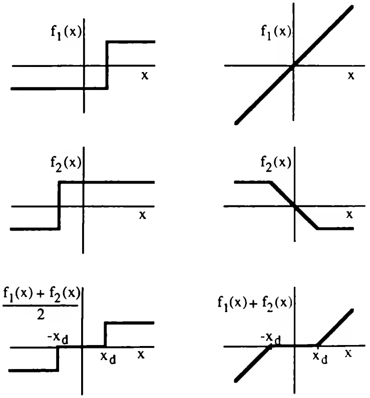
Consider next the derivation of the f_{ave} function for bang-bang control with dead zone, as illustrated in Figure 1b. Figure 2a shows how this control function can be represented as the superposition of the two bang-bang switch functions with inputs biased by $-x_d$ and x_d, respectively. It follows from Eq. (7) for the individual f_{avc} functions that the overall f_{ave} function in this case is given by

$$f_{avc} = \frac{|x_{n+1}-x_d| - |x_n-x_d| - |x_{n+1}+x_d| + |x_n+x_d|}{2(x_{n+1}-x_n)} \qquad (9)$$

Similarly, the linear function with dead zone illustrated in Figure 1d can be represented as the superposition of a linear function and an effort-limited linear function, as shown in Figure 2b. From Eq. (8) the following f_{avc} function is obtained:

$$f_{avc} = \frac{x_{n+1}+x_n}{2} - f_{LIM}(x_{n+i}, x_n) \qquad (10)$$

Here $f_{LIM}(x_{n+1}, x_n)$ is the f_{ave} function given earlier in Eq. (8) for effort-limited linear control.



a. Synthesis of bang-bang control with dead zone

b. Synthesis of linear control with dead zone

Figure 2. Synthesis of discontinuous control functions.

3. General Synthesis of Analytic Averaging Functions

From the example in Figure 2a it is evident that symmetric nonlinear functions in general with displacement discontinuities can be synthesized by a superposition of bang-bang switch functions, each with a respective weighting constant and input bias. Also, from the example in Figure 2b it is apparent that symmetric nonlinear functions with slope discontinuities can in general be synthesized by a superposition of linear functions with limiting, each with a respective weighting constant and input limit bias. Symmetric nonlinear functions consisting of both straight-line segments and displacement jumps can be synthesized by a superposition of both linear functions with limiting and bang-bang switch functions. In all cases the corresponding f_{avc} function can be written in terms of the f_{ave} functions given by Eqs. (7) and (8).

We next consider asymmetric nonlinear functions with displacement discontinuities. These can in general be synthesized by a superposition of step functions. The individual unit step function $d(x)$ in Figure 3a can be represented analytically by the formula $d(x) = (1+x/|x|)/2$. From Eq. (5) the corresponding f_{ave} function, which we denote as $D(x_{n+1}, x_n)$, is given by

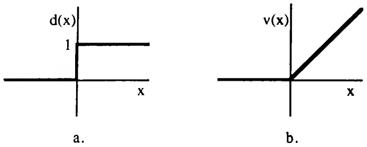


Figure 3. Unit step and ramp functions.

$$d_{ave} = D(x_{n+1},x_n) = \frac{1}{2} + \frac{|x_{n+1}|-|x_n|}{2(x_{n+1}-x_n)} \qquad (11)$$

Any staircase-type discontinuous function can be represented as the sum of biased step functions. Thus we can write

$$f(x) = f_0 + \sum_{i=1}^{N} k_i d(x-x_i) \qquad (12)$$

It follows from Eq. (11) that the corresponding f_{ave} function can be computed from the formula

$$f_{ave} = f_0 + \sum_{i=1}^{N} k_i D_i(x_{n+1}-x_i, x_n-x_i) \qquad (13)$$

Any asymmetric nonlinear function consisting of straight-line segments can be synthesized by a superposition of ramp functions. The individual unit ramp function $v(x)$ in Figure 3b can be represented by the formula $v(x) = (x+|x|)/2$. From Eq. (5) the corresponding f_{ave} function $V(x_{n+1}, x_n)$ is given by

$$f_{ave} = V(x_{n+1},x_n) = \frac{x_{n+1}(x_{n+1}+|x_{n+1}|) - x_n(x_n+|x_n|)}{4(x_{n+1}-x_n)} \qquad (14)$$

The general segmented function can be written as

$$f(x) = f_0 + m_0 x + \sum_{i=1}^{M} m_i v(x-x_i) \qquad (15)$$

Thus the corresponding f_{ave} function is computed from the formula

$$f_{ave} = f_0 + m_0\frac{x_{n+1}-x_n}{2} + \sum_{i=1}^{M} m_i V(x_{n+1}-x_i, x_n-x_i) \qquad (16)$$

Eqs. (11) through (16) can be built into a computer subroutine which will automatically calculate the $f_{ave}(x_{n+1},x_n)$ function given the data points defining any nonlinear function $f(x)$ that consists of linear segments plus displacement discontinuities.

4. Methods for determining x_{n+1}

It is clear from Eq. (5) that f_{ave} will be a function of x_{n+1} and x_n. When f_{ave} is computed during the nth integration frame, x_{n+1} may not be available. In this case it will be necessary to estimate x_{n+1}. In this section we consider several methods for making this estimate and the associated accuracy of the estimate. In the following sections specific examples will be used to illustrate several of the methods.

If x_n is a state variable or a known function of state variables, it may be possible to perform the state variable integrations during the nth frame prior to computing the f_{ave} function. In this case x_{n+1} will represent the most accurate estimate possible that is consistant with the algorithm being used for numerical integration. If it is not possible to obtain x_{n+1} in this fashion, then it must be estimated using some type of extrapolation algorithm.

Again, if x_n is a state variable, the time derivative $\dot{x}_n$ will be available for the estimate of x_{n+1}. If, alternatively, x_n is an analytic function of state variables, then $\dot{x}_n$ can be calculated, although for complex functions the calculation may not be trivial. In any event let us assume that $\dot{x}_n$ is available or can be computed. Then a first-order power series extrapolation formula for x_{n+1} is the following:

$$x_{n+1} = x_n + h\dot{x}_n \qquad (17)$$

From the Taylor series representation for x_{n+1} in terms of x_n it is easily seen that the error in x_{n+1} is approximately $-\ddot{x}_n h^2/2$. It is apparent that the extrapolation formula of Eq. (17) is identical with the Euler integration formula.

A second method for estimating x_{n+1} uses linear extrapolation based on x_n and x_{n-1}. Here the formula is

$$x_{n+1} = 2x_n - x_{n-1} \qquad (18)$$

In this case the error in x_{n+1} is approximately $-\ddot{x}_n h^2$, i.e., twice the error associated with the estimate of Eq. (17).

A second-order extrapolation method for x_{n+1} is based on x_n, x_{n+1} and $\dot{x}_n$. Here the formula is given by

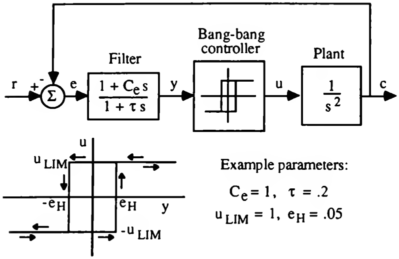
$$x_{n+1} = x_{n-1} + 2h\dot{x}_n \qquad (19)$$

Here the error in x_{n+1} is approximately $-\dddot{x}_n h^3/3$.

Other higher order extrapolation algorithms can of course be considered(6). Because of the asumption that dx/dt is constant in the derivation of the f_{ave} foumula, however, it is doubtful if much would be gained in going to more accurate extrapolation methods.

5. Bang-bang Control system Example

In this section we consider an example simulation of a dynamic system with discontinuities. Figure 4 shows a block diagram of the system, which consists of a pure inertia plant driven by a "bang-bang" controller with hysterisis. Proportional plus rate control is mechanized with the lead-lag filter shown in the figure. If the controller were also to include deadzone, the system would be representative of a typical spacecraft single-axis attitude control system.



Example parameters:

$C_e = 1,\ \tau = .2$

$u_{LIM} = 1,\ e_H = .05$

Figure 4. Bang-bang control system with hysterisis.

For the parameters shown in the figure the time response of the control system for zero input ($r = 0$) and two different initial conditions, $c(0)$, is shown in Figure 5. The other two states are initially zero. Note that the response approaches a limit cycle in each case. This is of course typical for any bang-bang control system.

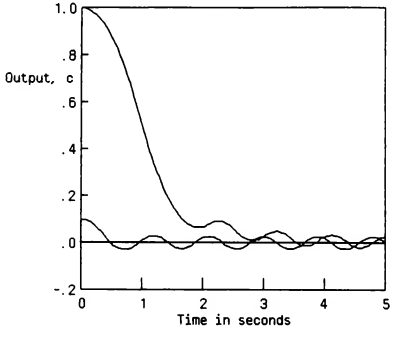


Figure 5. Transient response of control system for two different initial conditions.

The control system is described by the following state equations:

$$\dot{x} = \frac{1}{\tau}(r - c - x),\quad y = x + C_e\dot{x}\ ,$$
$$u = u_{LIM}\frac{y \pm e_H}{|y \pm e_H|}\quad ,\ \dot{c}_d = u,\quad \dot{c}\ = c \tag{20}$$

We consider first the simulation of the system using the second-order Adams-Bashforth (AB-2) predictor algorithm, which is probably the most popular real-time integration method. This results in the following difference equations when the standard method for nonlinear function evaluation is employed:

$$x_{n+1} = x_n + \frac{h}{2}(3fx_n - fx_{n-1}),\ fx_n = r_n - c_n - x_n$$
$$y_n = x_n + C_e fx_n,\ S_n = \frac{y_n + e_H S_{n-1}}{|y_n + e_H S_{n-1}|}\ ,\ u_n = u_{LIM}S_n \tag{21}$$
$$c_{d_{n+1}} = c_{d_n} + \frac{h}{2}(3u_n - u_{n-1}),\ c_{n+1} = c_n + \frac{h}{2}(3c_{d_n} - c_{d_{n-1}})$$

Here we have introduced a discrete state variable $S_n = \pm 1$ in order to mechanize the hysterisis bias e_H, where the polarity of the bias depends on the previous switch state, S_{n-1}.

Next we replace the conventional evaluation of the bang-bang control function u_n with the f_{ave} function in accordance with Eq. (7). The formula for the switch variable S_n given in Eq. (21) is still retained to preserve the evaluation of the hysterisis bias $\pm e_H$. But the equations for u_n and $c_{d_{n+1}}$ are replaced by

$$\tilde{u}_{n+1/2} = u_{LIM}\frac{|y_{n+1} + e_H S_{n-1}| - |y_n + e_H S_{n-1}|}{y_{n+1} - y_n}\ , \tag{22}$$

$$c_{d_{n+1}} = c_{d_n} + h\tilde{u}_{n+1/2} \tag{23}$$

In Eq. (22) we have denoted the f_{ave} function by $\tilde{u}_{n+1/2}$. This is because the average value of the bang-bang control over the interval from nh to (n+1)h is equivalent to an estimate of u halfway through the interval as it is used in the integration algorithm of Eq. (23).

In Figure 6 the error in the simulated control system output c when using AB-2 integration for $c(0) = 1$ is plotted as a function of time. Two cases are shown. In one case AB-2 is used with the standard method of bang-bang function evaluation, as represented by the difference equations in (21). In the second case the function averaging method is used to represent and integrate the bang-bang control, as reflected by Eqs. (22) and (23). Use of the averaging method has clearly made a very significant improvement in the accuracy of the solution. It should be noted that in both cases the AB-2 algorithms were started so that there was no error for the initial 1 second portion of the solution, over which the controller output u is equal to -1. It is when u switches from -1 to +1 and when subsuquent switches in u occur that the error transients are generated.

Eq. (23) can be viewed as a modified form of Euler integration wherein the half integer state variable derivative at n+1/2 instead of the derivative at the integer n is utilized in computing the n+1 state. This form of modified Euler integration can actually be used for many if not all integrations in

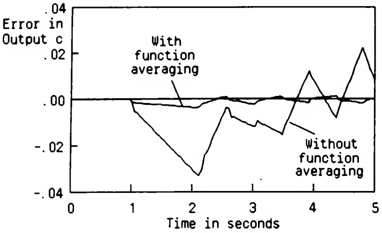


Figure 6. Error in control system output when using AB-2 integration.

a simulation. It has the advantage that the dynamic errors associated with the method are proportional to $h^2/24$, compared with $5h^2/12$ for AB-2 integration$^{(7)}$. For this reason and because of its compatibility with the function averaging method for handling discontinuities, we next consider the modified Euler method for all integrations in the simulation of our bang-bang control system.

When applying modified Euler integration it is necessary to designate some of the state variables at half integer rather than integer time steps. For the control system example here we choose to represent the output position c at integer steps, with the output velocity c_d and the filter state x represented at half-integer steps. When the standard method for nonlinear function evaluation is used, the modified Euler difference equations for simulating the control system state equations in (20) become the following:

$$x_{n+1/2} = A_x x_{n+1/2} + B_x(r_n - c_n)\,, \quad A_x = \frac{1 - \frac{h}{2\tau}}{1 + \frac{h}{2\tau}}\,, \quad B_x = \frac{\frac{h}{\tau}}{1 + \frac{h}{2\tau}} \tag{24}$$

$$y_n = \tilde{x}_n + \frac{C_e}{\tau}(r_n - c_n - \tilde{x}_n)\,, \quad \tilde{x}_n = \frac{x_{n+1/2} + x_{n-1/2}}{2} \tag{25}$$

$$S_n = \frac{y_n + e_{II} S_{n-1}}{|y_n + e_{II} S_{n-1}|}\,, \quad u_n = u_{LIM} S_n \tag{26}$$

$$c_{d_{n+1/2}} = c_{d_{n-1/2}} + h u_n\,, \quad c_{n+1} = c_n + h c_{d_{n+1/2}} \tag{27}$$

We note in Eq. (24) that modified Euler has been used to integrate $(r_n - c_n - \tilde{x}_n)/\tau$ and thus obtain $x_{n+1/2}$ from $x_{n-1/2}$. Here the estimate $\tilde{x}_n$ is obtained by averaging $x_{n+1/2}$ and $x_{n-1/2}$, as shown in Eq. (25). In this case we are in effect using trapezoidal integration for the state x. The resulting difference equation is solved explicitly for $x_{n+1/2}$, which leads to Eq. (24) with the coefficients A_x and B_x. The estimate $\tilde{x}_n$ is also used in Eq. (25) for the filter output y_n, which in turn is used in Eq. (26) to compute the bang-bang switch output S_n and hence the controller output u_n. In Eq. (27) the velocity $c_{d_{n+1/2}}$ is computed from $c_{d_{n-1/2}}$ using u_n with modified Euler integration, as is c_{n+1} from c_n using $c_{d_{n+1/2}}$.

To employ the function averaging method with modified Euler integration we utilize the following equations for the calculation of $\tilde{u}_n$, the average value of u over the interval from $(n-1/2)h$ to $(n+1/2)h$:

$$\tilde{u}_n = u_{LIM} \frac{|\tilde{y}_{n+1/2} + e_H S_{n-1}| - |\tilde{y}_{n-1/2} + e_H S_{n-1}|}{\tilde{y}_{n+1/2} - \tilde{y}_{n-1/2}} \tag{28}$$

where

$$\tilde{y}_{n+1/2} = \frac{3}{2} y_n - \frac{1}{2} y_{n-1}\,, \quad \tilde{y}_{n-1/2} = \frac{y_n + y_{n-1}}{2} \tag{29}$$

In Eq. (28) for $\tilde{u}_n$ the formula is based on estimates $\tilde{y}_{n+1/2}$ and $\tilde{y}_{n-1/2}$, as obtained in Eq.(29) from y_n and y_{n-1} using first-order extrapolation and interpolation, respectively. To use modified Euler integration with function averaging for the bang-bang control, then, $\tilde{u}_n$ is employed instead of u_n in Eq. (27).

In Figure 7 the error in simulated control system output c when using modified Euler integration for $c(0) = 1$ is plotted versus time. As before, two cases are shown, one without the averaging method and the second using the averaging method, as mechanized with Eqs. (28) and (29). Again we note the very substantial accuracy improvement when using the function averaging algorithm.

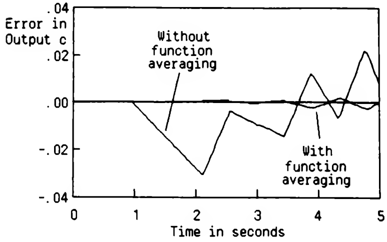


Figure 7. Error in control system output when using modified Euler integration.

Although the accuracies observed in Figure 6 for AB-2 integration are comparable with those in Figure 7 for modified Euler integration, the modified Euler has some advantags over the AB-2 implementation. In particular, the AB-2 mechanization in Eq. (22) requires y_{n+1} and therefore r_{n+1} for the nth integration frame. If r is a real-time input, r_{n+1} will not be available, although it could be estimated from r_n and r_{n-1} by extrapolating ahead by h seconds. On the other hand, in the modified Euler mechanization in Eqs. (28) and (29) y_{n+1} and hence r_{n+1} is not required. Thus the simulation will be compatible with real-time inputs. It is true in this case that extrapolation in Eq. (29) is required in computing $\tilde{y}_{n+1/2}$, but it is extrapolation over the interval $h/2$ rather than h. This reduces the extrapolation error by a factor of four. We might also note that when the lead-lag filter has a second-order lag, i.e., a transfer operator given by $(1+C_e s)/(1+\tau s)^2$, the modified Euler algorithm permits the calculation of $y_{n+1/2}$ without the use of extrapolation.

We have also noted earlier in this section that the modified Euler method in general has a dynamic accuracy which is an order of magnitude better than AB-2 (error coefficient proportional to $h^2/24$ compared with $5h^2/12$). In the bang-bang control system example used here the dynamic errors are dominated by the errors associated with the discontinuous control function. Indeed, this is why the function averaging method made such a dramatic improvement in the accuracy. Because of this, however, the poorer overall accuracy of AB-2 is not so evident, especially when we note that AB-2 produces an exact result when simulating our pure inertia plant with a constant input. In more complex problems the authors have found that the modified Euler method enjoys a substantial accuracy advantage over AB-2[7].

Finally, there are fewer startup problems associated with modified Euler integration. In particular, the initial integration step for the half integer states $x_{n+1/2}$ and $cd_{n+1/2}$ starting with x_0 and cd_0 is taken as $h/2$ rather than h. Normally AB-2 is started with Euler integration for the first step, which introduces a substantial error transient. To obtain the accuracy shown in Figure 6 we were forced to compute startup deriva-tives at $t = -h$. In general this may be inconvenient.

Previous studies of the function averaging technique described in this paper have also shown impressive accuracy improvement when the method is used in handling slope discontinuities, e.g., for effort-limited linear controllers, as well as the displacement discontinuities of bang-bang controllers[3]. These studies have demonstrated that the function averaging technique can be used successfully in conjunction with other integration methods such as AB-3, AB-4, RK-2 and RK-4. The previous studies have also shown that the errors resulting from both types of discontinuities are strongly dependent on the time at which the discontinuity occurs during the interval between nh and $(n+1)h$. It follows that small changes in the initial conditions in the example considered here will make substantial changes in the size of the error transients, especially for the cases where the function averaging method is not used. Hence the results in Figures 6 and 7 should only be considered as typical, not all encompassing.

6. Conclusions

The use of an analytic averaging technique improves considerably the accuracy in simulating dynamic systems with discontinuous state-variable derivatives. The method is especially effective for fixed integration step sizes, such as are used in real-time simulation. The improved accuracy has been demonstrated in the simulation of a bang-bang control system with hysterisis for both AB-2 integration and a modified form of Euler integration. The analytic averaging formulas for applying the method to any nonlinear function consisting of linear segments and displacement discontinuities have been presented.

7. References

1. Carver, M.B., "Efficient Integration over Discontinuities in Ordinary Differential Equations," *Mathematics and Computers in Simulation*, vol. xx, no. 3, pp 190-196, 1978.

2. Ellison, D., "Efficient Automatic Integration of Ordinary Differential Equations with Discontinuities," *Mathematics and Computers in Simulation*, vol. xxiii, no. 1, pp 12-20, 1981.

3. Halin, H.J., "Integration Across Discontinuities in Ordinary Differential Equations Using Power Series," *Simulation*, vol. 32, no. 2, pp 33-45, 1979.

4. Howe, R.M., X.A. Ye and B.H. Li, "An Improved Method for Simulation of Dynamic Systems with Discontinuous Nonlinearities," *Trans. of the Society for Computer Simulation*, Vol.2, No. 1, pp 169-187, 1985.

5. Howe, R.M., "A New Method for Handling Discontinuous Nonlinear Functions in Digital Simulation of Dynamic Systems," *Proc 1978 Summer Computer Simulation Conference*, pp 72-79, The Society for Computer Simulation, P.O. Box 17900, San Diego, CA 92117.

6. Howe, R.M., "Dynamics of Digital Extrapolation and Interpolation," *Trans. of the Society for Computer Simulation*, Vol.2, No. 1, pp 169-187, 1985.

7. Howe, R.M., "The Role of Modified Euler Integration in Real-time Simulation," *Proceedings of the Conference on Aerospace Simulation III*, San Diego, 1988; pp 277-285. The Society for Computer Simulation, P.O. Box 17900, San Diego, CA 92117.

CONVERSION OF EXISTING FORTRAN SIMULATION PROGRAMS TO A GENERAL PURPOSE SIMULATION LANGUAGE

88-4581-CP

Steve Zammit
Applied Dynamics International
Ann Arbor, Michigan

1 Abstract

This paper discusses guidelines for converting FORTRAN based simulation programs to a general-purpose simulation language, such as ADSIM, the Applied Dynamics International Simulation Language. This process normally involves developing an understanding of the FORTRAN model, developing the simulation math model, converting the model, debugging the converted model, and understanding the differences which may occur between the two simulation programs. Finally, an example is presented that converts a FORTRAN program to ADSIM.

2 Introduction

Many times an engineer is faced with the task of converting an existing FORTRAN-based simulation program to any one of the commercially available simulation languages. These languages include ACSL (Advanced Continuous Simulation Language), CSSL (Continuous System Simulation Language), and ADSIM (Applied Dynamics International Simulation Language).

The reason for converting the program may be in an effort to standardize in-house simulation programs or, perhaps, to execute the simulation on a special-purpose digital computer. An example of a special-purpose digital computer is the Applied Dynamics International AD 100, which is ideally suited for real-time, hardware-in-the-loop simulation.

The FORTRAN-based simulation program may be the result of many years of development by any number of different programmers. Thus, documentation of the program internals may be sketchy, at best. Different programming style and the use of inconsistent programming techniques also complicate understanding the program flow. Mathematical models upon which the FORTRAN program is based may be incomplete or altogether nonexistent.

While the authors of the simulation languages have paid a great deal of attention to proper numerical techniques, the authors of the FORTRAN-based simulation may have made errors with respect to some of the more subtle numerical issues. These issues include proper numerical integration, the limiting of state variables, and sorting of the dynamic equations.

This paper discusses a general procedure for the conversion process. The first step of the conversion normally involves the derivation of the model physics from the FORTRAN simulation source code. This step serves a dual purpose: First, it provides documentation for the FORTRAN-based simulation; Second, it provides an understanding of the model needed in order to rewrite the simulation in any of the standard simulation languages. Due to the evolved nature of most of the large FORTRAN simulation programs, this step of the process, more often than not, uncovers some questionable modeling techniques.

After determining the model physics from the FORTRAN simulation, the engineer/programmer must make some decisions with respect to recoding the model in the chosen simulation language. Most of the languages have built-in libraries for traditional types of mathematical procedures such as function generation, dead zone, hysteresis, limiting functions, etc. In order to create the most efficient code possible, these language elements should be used as much as possible. FORTRAN goto statements should be replaced with structured programming constructs in order to create more readable and maintainable simulation code.

Finally, an example is presented that shows the steps taken to convert an existing FORTRAN-based simulation program into ADSIM, the simulation language for the Applied Dynamics International AD 100 simulation computer.

3 The FORTRAN-Based Simulation Model

Certainly the first step in the conversion process is to develop a thorough understanding of the FORTRAN based simulation program. The ability to run the program and obtain numerical and graphical results will also aid the debugging phase of the project. Using a FORTRAN symbolic debugger, such as the one commercially available with the VAX/VMS operating system, will allow one access to any variable present in the FORTRAN simulation.

Since no two FORTRAN-based simulation programs are completely alike, we must limit our discussion to a structure that most frequently occurs. Normally, this structure is based upon subroutines that are linked together via a FORTRAN common area. The common area is used to communicate data values among the various subroutines of the simulation.

Viewing this subroutine structure on a macroscopic level, it should be possible to detect the simulation derivative evaluation section(s), as well as the integration loop. The numerical integration method or methods used should be identified, and, if at all possible, these methods should be incorporated into the real-time simulation. Sometimes integrations are found to be embedded in the derivative evaluation section of the FORTRAN simulation. These should be pulled out as all simulation derivatives should be integrated and updated at the same point during the simulation frame. The general framework that should be employed for continuous system simulation is listed below.

```
Initialize State Variables
While $SystemTime < EndTime$ Do
    Evaluate Simulation Derivatives
    Integrate Simulation Derivatives
    $SystemTime = SystemTime + StepTime$
EndDo
```

Another situation that frequently occurs with FORTRAN simulation code is the improper sorting of the dynamic equations. The simple rule here is that no variable should be used on the right-hand side of an equation without first having been calculated. To do otherwise would be to take a one frame delay in the variables that are calculated out of order. Tolerating this one-frame delay is similar to introducing a simple lag into the system with a time constant equal to the integration step size. As the simulation language will properly sort the dynamic equations, the computational order of the equations of motion will not be preserved. This will certainly have an effect on the numerical solution of the real-time simulation. However, the simulation language solution will be *correct*, whereas the FORTRAN-based simulation is in *error*. These occurrences should be noted as they will become important during the validation phase of the real-time model.

Another manifestation of sorting the dynamic equations is circular equations. This condition occurs when two or more equations are interdependent. One possible solution is to try to reformulate the dynamic equations. Another solution is to simply take a one-frame delay in one or more of the variables in order to break the loop. Again, the latter solution has its ramifications and is certainly less desirable.

Function generation subroutines should be examined for the type of interpolation scheme employed. Normally, linear interpolation between the breakpoint values is used. Of more interest is what the FORTRAN program does for values of the independent variables *outside* the range of the breakpoint table. For example, does the function generation routine hold the value of the function constant outside the defined range of the breakpoint table, or does the routine perform a linear extrapolation?

Frequently, in the case of missile and aircraft simulation, extremely large function generation tables may be used to accurately represent the aerodynamic coefficients. The FORTRAN-based simulation program usually reads these function values from an ASCII data file into the simulation prior to the run. As an example, consider the following function for a lift coefficient.

$$c_l = f(alpha, mach, alt)$$

Assume that angle of attack, *alpha*, has 17 breakpoints, the Mach number, *mach*, has 11 breakpoints and the altitude, *alt*, has 65 breakpoints. This gives a total of 12,155 function values for c_l. In FORTRAN these function values could be stored in a three- dimensional array of size (17, 11, 65). The question to be answered by the simulationist is how the data is ordered in the data file for multivariable functions. Usually, this is easily determined by looking at the FORTRAN simulation read statement(s) for the function data and determining which variable *varies most rapidly*. If FORTRAN data statements are used recall that the left-most variable in a multidimensional array varies most rapidly. Independent variables or *breakpoint tables* must be correlated with the analogous simulation variables.

4 Math Model Development

In an effort to achieve simulation frame times that accurately integrate dynamic systems in real time, many simulation languages require that the dynamic equations be written in scalar form. The simple reason for this is that array notation takes additional time for address-cell arithmetic. Thus, if at all possible, the math model should be written in scalar form.

The integration method(s) used by the FORTRAN-based model must also be examined closely. Determine which method or methods are being used. If it is one of the single-step Adams-Bashforth integration methods, determine what type of start up procedure is being used. Recall that the Adams-Bashforth methods use the current as well as the *past* derivatives to predict the next value of the state variable. It may be that at time equals zero, or at the start time of the simulation, these values are unknown. In fact, this is normally the situation unless the system is at a steady state or trim condition, in which case, the derivatives are very small, i.e., negligible. The Adam-Bashforth methods are well suited for real-time simulation, since they require external inputs only for the current point in time.

On the other hand, multistep methods, such as the traditional Runge-Kutta methods, do not have a start-up problem. However, these methods are not at all suited for real-time simulation as they require inputs to the integration method before the inputs become available. In addition, these methods produce new state variables two or four times slower (dependent upon the Runge-Kutta method used) than their single step counterparts.

Structured programming, such as if-then FORTRAN blocks, should be used as opposed to goto statements. FORTRAN IV did not support if-then blocks and thus many of the older FORTRAN-based simulations are full of goto statements. Trying to follow the path of the simulation through hundreds of goto statements can make one crazy; sometimes it is easier to just let the program tell *you* where *it* is going. This can easily be accomplished by using the FORTRAN debugger or by simply inserting print statements in the paths of concern. Using if-then blocks will go a long way toward making the real-time simulation code more readable as well as maintainable.

5 The Simulation Language

It should almost go without saying that an important factor for any engineer or engineering manager to consider is the selection of a simulation language that best suits the company's short-term and long-term needs. A sampling of questions that should be answered are:

- Is the language flexible enough to satisfy the wide range of engineering applications most simulation laboratories encounter?
- Can the language be executed on different types of hardware systems to meet these varied needs?

- Are the low-level macros and simulation executives easily modified in order to simulate the *unique* and *complex* systems your laboratory will encounter?

- For real-time simulations, is the hardware/software *package* fast enough to accurately integrate your highest frequencies?

- Are the hardware resources large enough to accomodate your largest simulation program (i.e., in terms of program memory and function data memory)?

- Is the hardware/software system easily expanded to include analog and digital communication in real time (hardware-in-the-loop simulation)?

In order to answer the questions above and make a choice among simulation languages, a significant amount of training in the languages must be carried out. After a selection is made, it must be assumed that the engineer understands the simulation language of his choice. This understanding must include the more subtle numerical issues embedded within the language, for these will certainly pose the most persistent problems during the verification and validation phase of the conversion project.

6 Program Conversion

To demonstrate a simple conversion from FORTRAN to ADSIM, a closed-loop third-order system is used. The transfer functions for the system are shown in the figure 1.

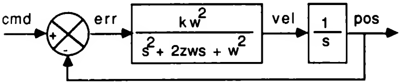


Figure 1: Third-Order Closed-Loop System

The FORTRAN main program is listed below. The FORTRAN include statement brings in a file containing array declarations, common- area definitions, and equivalence statements. This file is simply included in the FORTRAN main program and in the subroutines that are called from the main program. In terms of execution time and program clarity, this method is of sharing data is far better than passing data via subroutine invocations.

After the include statement, simulation parameters are assigned constant values through FORTRAN data statements. These terms include initial condition values for state variables, simulation run specifications, and system parameters.

The prerun portion of the code comes next. Here, the state variables are assigned their initial conditions, and the simulation time and the simulation frame counter are set equal to zero.

Next, the system clock is initialized for the purpose of measuring the total execution time of the dynamic portion of the simulation. For a simulation in real time, the time returned by the system clock at the end of the simulation will exactly equal the end time of the simulation.

Then comes the dynamic portion of the simulation. In the dynamic loop, derivatives and integrations are performed until the variable systemtime becomes greater than the simulation variable testendtime. When this occurs, the simulation ceases, and the clock is read for the total execution time. Note that all output statements have been removed for clarity.

```
      program actuator
      include 'common.for'
* Constants.
      data acc0     /0.0d0/
      data vel0     /0.0d0/
      data pos0     /0.0d0/
      data steptime /21.3000002986519D-06/
      data endtime  /0.1d0/
      data cmd      /0.70d0/
      data k        /280.0d0/
      data z        /0.60d0/
      data wn       /450.0d0/

* Initialize time and frame counter.
      systemtime = 0.0d0
      framecount = 0.0d0
* Initialize state variables.
      pos = pos0
      vel = vel0
      acc = acc0
      testendtime = endtime-0.50d0*steptime
      delta = secnds(0.0)
* Simulation dynamic loop.
10    call derivative
      if (systemtime.ge.testendtime) goto 20
      call integrate
      systemtime = systemtime + steptime
      framecount = framecount + 1.0d0
      goto 10
20    time = secnds(delta)
      end
```

The following is a listing of the algebraic and derivative evaluation subroutine. It simply contains the single equation for the error signal and the three derivatives of the third-order system displayed in figure 1. Again, all constants, states, derivatives, etc., are tranferred to other subroutines via common areas as defined in the file *common.for*.

```
      subroutine derivative
      include 'common.for'
* Evaluate algebraic and derivative terms.
      err  =  cmd-pos
      accp =
     .-2.0d0*z*wn*acc-wn*wn*vel+k*wn*wn*err
      velp =  acc
      posp =  vel
      return
      end
```

The following is the integration subroutine. The integration routine selected is the Adams-Bashforth second-order method (AB-2). While this method is suitable for real-time simulation, it has a start-up problem in the sense that AB-2 needs the current derivative as well as the the derivative calculated at frame $n-1$. To get around this problem, AB-1 is selected for the first frame since it needs only the derivative at frame n. After the first frame has completed the subsequent frames are integrated with AB-2. Note that this routine also shifts the derivatives for the next frame's integration pass.

```
      subroutine integrate
      include 'common.for'
* Integrate with AB-1 at t=0; then use AB-2.
      if (systemtime.eq.0.0d0) then
       acc = acc+steptime*accp
       vel = vel+steptime*velp
       pos = pos+steptime*posp
      else
       acc =
     . acc+0.50d0*steptime*(3.0d0*accp-accpnm1)
       vel =
     . vel+0.50d0*steptime*(3.0d0*velp-velpnm1)
       pos =
     . pos+0.50d0*steptime*(3.0d0*posp-pospnm1)
      endif
* Shift derivatives.
      accpnm1 = accp
      velpnm1 = velp
      pospnm1 = posp
      return
      end
```

For completeness, the file *common.for* is listed below.

```
      implicit real*8(a-z)
      real*4 delta
      real*4 time
      real*8 constants   (10)
      real*8 derivatives (10)

      real*8 states      (10)
      real*8 runspecs    (10)
      real*8 algebraics  (10)
      common /constantscmn/  constants
      common /derivativecmn/ derivatives
      common /statecmn/      states
      common /runspecscmn/   runspecs
      common /algebraiccmn/  algebraics
      equivalence (constants(1),k),
     .            (constants(2),z),
     .            (constants(3),wn),
     .            (constants(4),cmd),
     .            (constants(5),pos0),
     .            (constants(6),vel0),
     .            (constants(7),acc0),
     .            (constants(8),maxpos),
     .            (constants(9),cr)
      equivalence (derivatives(1),accp),
     .            (derivatives(2),velp),
     .            (derivatives(3),posp)
      equivalence (states(1),acc),
     .            (states(2),vel),
     .            (states(3),pos)
      equivalence (runspecs(1),steptime),
     .            (runspecs(2),endtime)
      equivalence (algebraics(1),err),
     .            (algebraics(2),systemtime),
                  (algebraics(3),framecount)
```

Now the task at hand is to convert the FORTRAN program into ADSIM. ADSIM is a simulation language that can be used either on the Applied Dynamics AD 100 high-speed digital computer or on one of the VAX family of general-purpose computers.

ADSIM is a block-structured language, similar to other simulation languages. One of these blocks is denoted as the *dynamic block* and it simply contains the equations that are to be solved recursively in time. Time derivatives are denoted by the prime symbol, as shown in the dynamic block in the following ADSIM source code listing. These derivatives are automatically integrated for the corresponding state variables by the simulation executive for continuous system simulation. This executive is invoked via the ADSIM statement *execute continuous*. System variables embedded in the simulation also keep track of system time and frame count, to name a couple.

ADSIM also supports other simulation executives. A simulation executive for multiframe-rate integration will soon be supplied as a normal part of the software release. Because the simulation executive is also written in ADSIM, it is readily modified to support other types of environments such as mixed-data systems, systems with multiple dynamic portions, etc.

Initial conditions on state variables are set by using the state variable name with an @ symbol appended to the end of the name. Constants are assigned default values via ADSIM data statements. All values can be changed at run time interactively without recompilation.

The complete converted program is listed below.

```
TITLE Third-order closed-loop system.
! Simulation dynamics:
DYNAMIC continuous
  pos' = vel
  vel' = acc
  acc' = -2.0*z*wn*acc-wn*wn*vel+k*wn*wn*err
  err  = cmd-pos
END DYNAMIC continuous
! Assign initial conditions to state variables:
DATA pos@ = 0.0, vel@ = 0.0, acc@=0.0
! Assign default values to actuator parameters:
DATA z = 0.6, wn = 450.0, cmd = 0.7, k = 280.0
! Executive for continuous system simulation:
EXECUTE continuous
```

7 Debugging the Converted Program

First, making the assumption that the FORTRAN based model is the simulation truth model, the initial step of the debugging process is to obtain data from the FORTRAN model by performing a static check. A static check performs no integrations, rather, derivatives are evaluated once using the initial conditions on the state variables.

The following is a listing of a log file made by using the VAX FORTRAN debugger. A break point is set at the return statement in subroutine derivative and the code is then executed. After hitting the breakpoint the values for the state variables and derivatives are displayed.

```
DBG> set break/return derivative
DBG> go
%DEBUG-I-DYNMODSET, setting module DERIVATIVE
break on return from routine DERIVATIVE at
DERIVATIVE\%LINE 53
    53:       return
DBG> examine pos,vel,acc,posp,velp,accp,err
DERIVATIVE\POS:  0.0000000000000000
DERIVATIVE\VEL:  0.0000000000000000
DERIVATIVE\ACC:  0.0000000000000000
DERIVATIVE\POSP: 0.0000000000000000
DERIVATIVE\VELP: 0.0000000000000000
DERIVATIVE\ACCP: 39690000.00000000
DERIVATIVE\ERR:  0.7000000000000000
DBG> exit
```

Now the ADSIM model is executed on the AD 100. The ADSIM model contains a built in interactive utility named INTERACT. INTERACT provides a control environment for running the simulation in several different run modes, namely, static check, single, continuous, and repetitive run modes. The AD 100 with INTERACT also provides a default environment of real-time simulation. Note that the frame time for this example is 21.3 microseconds. Roughly 17.5 microseconds of this time is overhead imposed by the simulation executive and is a constant value, independent of program size. Again, all variables in the simulation may be changed via INTERACT without recompiling.

Thus, a static check is performed simply by issuing the INTERACT command *go check*. After the AD 100 suspends, the states, derivatives and simulation algebraic terms are displayed. A quick check shows excellent agreement between the FORTRAN model and the ADSIM model.

```
$ run act
MPS10 Interact V5.1   9 September 1986
ADSIM Interact V5.21 15 December 1987
(C) Copyright 1983,1987
    by Applied Dynamics International, Inc.
    All Rights Reserved
AD100> start 0 act
AD 100 console 0 attached
The AD 100 system file is loaded
The IOCP system file is loaded
Created: 17-JUL-88 14:01:03 ADSIM V5.21
Third order closed loop system.
Revised:
EXECUTE continuous  V5.21b  17 Dec 1987
No graphics options installed
Single runmode, realtime environment.
frametime       21.3000E-6
steptime        21.3000E-6
speedup          1.0000
endtime          1.0000

Type GO HELP for some helpful information.
AD100> format 13
AD100> go check
Check runmode, realtime environment.
The AD 100 is suspended.
Suspended solution for static/step check.
AD100> data/der/state/alg
DERIVATIVE AND NEXT-STATE variables
acc'                        39.6900000000000E+6
vel'                         0.0000000000000
pos'                         0.0000000000000
STATE variables
acc                          0.0000000000000
vel                          0.0000000000000
pos                          0.0000000000000
ALGEBRAIC variables
step_time                   21.3000002986519E-6
frame_time                  21.3000002986519E-6
speed_up                     1.0000000000000
end_time                     1.0000000000000
system_time                  0.0000000000000
cmd                        699.9999999998181E-3
err                        699.9999999998181E-3
k                          280.0000000000000
wn                         450.0000000000000
z                          600.0000000003638E-3
```

Next, the same variables are checked at the end of the FORTRAN simulation. Again, the VAX debugger is used to log the state and derivative data values. A log file of the debugger output follows.

```
DBG> go
%DEBUG-I-EXITSTATUS, is '%SYSTEM-S-NORMAL,
normal successful completion'
DBG> examine pos,vel,acc,posp,velp,accp
ACTUATOR\POS:   0.6999615751683599
ACTUATOR\VEL:   -0.1281003220806408
ACTUATOR\ACC:   24.47985658984892
ACTUATOR\POSP:  -0.1281003220806408
ACTUATOR\VELP:  24.47985658984892
ACTUATOR\ACCP:  14899.88061680496
DBG> exit
```

And finally, the AD 100 is started in single run mode for a 0.1 second simulation. After the AD 100 halts, the states, derivatives, and algebraics are displayed. Comparing this set of values to the values obtained from the FORTRAN model, it is found that the relative errors between the two data sets are roughly of the order of 10e-09 or better.

```
AD100> runspecs endtime 0.1
AD100> go single
Single runmode, realtime environment.
The AD 100 is running.
```

```
AD100>
The AD 100 is halted.
AD100> data/der/state/alg
DERIVATIVE AND NEXT-STATE variables
acc'                       14.8998806089163E+3
vel'                       24.4798565895180
pos'                     -128.1003220876755E-3
STATE variables
acc                        24.4798565895180
vel                      -128.1003220876755E-3
pos                       699.9615751683450E-3
ALGEBRAIC variables
step_time                  21.3000002986519E-6
frame_time                 21.3000002986519E-6
speed_up                    1.0000000000000
end_time                  100.0000000000227E-3
system_time               100.0035014021705E-3
cmd                       699.9999999998181E-3
err                        38.4248314730939E-6
k                         280.0000000000000
wn                        450.0000000000000
z                         600.0000000003638E-3
AD100> exit
```

Various researchers have estimated that the verification and validation portion of a simulation can consume between 30 to 60 percent of a particular project's schedule and budget.[1] The interactive nature of ADSIM with the INTERACT utility can make the task of verification and validation much easier and allow one to develop a *feel* for the model being simulated.

Time histories of position and velocity for the FORTRAN simulation and the ADSIM simulation have been included in figures 2 and 3, respectively. The FORTRAN output plots were created by saving data into a file and then running a post-process graphics package. The ADSIM plots were made by INTERACT, from data collected in real time. Both sets of plots were created on a DEC VaxStation.

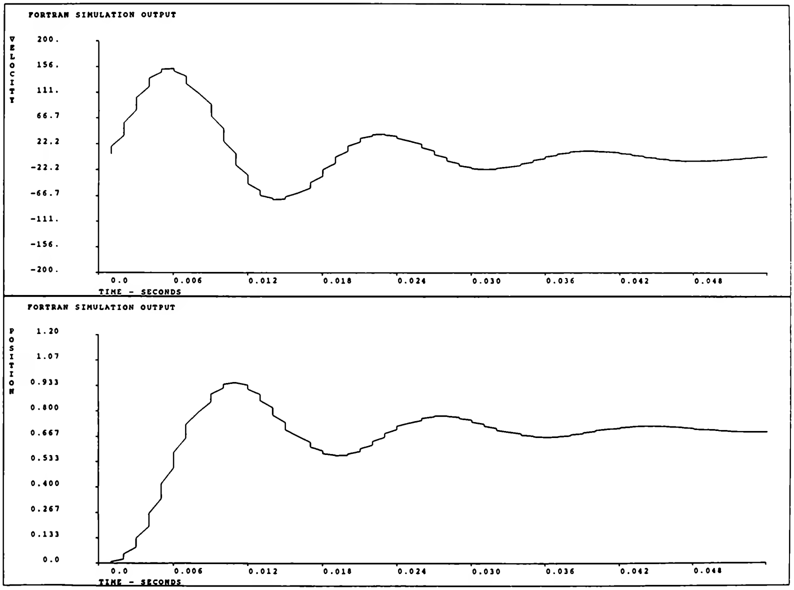


Figure 2: FORTRAN Simulation Times Histories

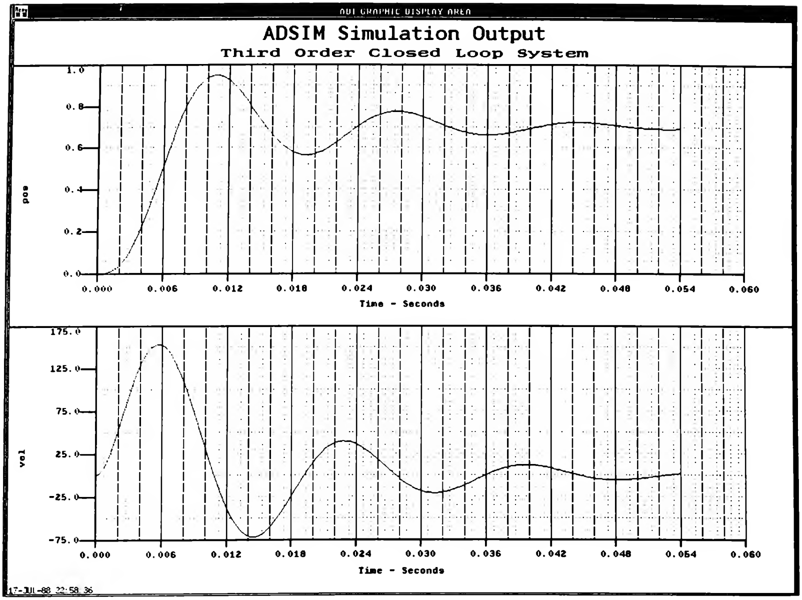


Figure 3: ADSIM Simulation Times Histories

8 Conclusions

Several issues regarding the potential problems encountered during a conversion from a FORTRAN-based simulation model to a general purpose simulation language have been discussed. A general procedure for the conversion process has been proposed.

A third-order closed-loop FORTRAN model has been converted into ADSIM for execution on the AD 100 high-speed digital computer. The ADSIM model running on the AD 100 had a frame time of 21.3 microseconds, 17.5 of which was the overhead for the simulation executive. Thus, the execution time for the model alone was 3.8 microseconds. The AD 100 model executed the model in real time by default, and thus the overall simulation time was exactly equal to the simulation end time of 0.1 seconds.

On the other hand the FORTRAN model running on a single user DEC microVax posted an overall execution time 1.054688 seconds, with identical parameters for integration step size and simulation end time. Thus the microVax was over 10.5 times slower than real time.

9 References

1. Continuous System Simulation Language (Version 4), Simulation Services, Nilsen Associates.

2. ADSIM Reference Manual (Version 5.21), Applied Dynamics International

3. Programming in VAX FORTRAN (Version 4), Digital Equipment Corporation

REAL-TIME SIMULATION OF HELICOPTERS USING THE BLADE ELEMENT METHOD

88-4582-CP

Koos Zwaanenburg
Applied Dynamics International
Ann Arbor, Michigan

1 ABSTRACT

This paper discusses the experience gained in applying the AD 100 computer to the real-time simulation of helicopters using the blade element method. The use of a single computer, together with the ADSIM simulation language, eliminates many of the problems associated with the application of multiple general-purpose computers, programmed in FORTRAN, to such a large and complex real-time simulation. In particular, this paper shows that the implementation of the blade element rotor equations for the UH-60A Black Hawk helicopter is a straightforward task on the AD 100.

2 INTRODUCTION

For the main rotor, most helicopter flight simulators for pilot training have, until now, used simplified math models, which permit a general-purpose digital computer to solve the equations in real time. Even in the case of real-time helicopter simulators for engineering purposes, the simplified main rotor models have often been used. Because of an ever-increasing emphasis on improved dynamic fidelity, however, considerable interest has developed in the use of the blade element method for modeling the main rotor in real-time helicopter simulations. This is a very computationally intensive task requiring much faster computers than those needed for the simplified main rotor models. A number of methods for achieving the required speeds have been proposed and in some cases implemented, including the use of specially configured hybrid computers, many microprocessors connected in parallel, the interconnection of a number of high-speed general-purpose computers, and the use of supercomputers. The use of multiple general-purpose computers programmed in FORTRAN, with the attendant problems in real-time synchronization, timing, and I/O, presents especially challenging problems. In this paper, an alternative approach is described, namely the use of a single, high-performance computer with an architecture optimized for solving non linear ordinary differential equations. The computer is the AD 100, which has been used extensively for real-time simulation of high-performance missiles, spacecraft, and aircraft but which only recently has been considered for helicopter simulation.

3 MODEL DESCRIPTION

Developed at Sikorsky,[1] the math model used in this paper describes a flying-qualities analysis model of the UH-60A Black Hawk helicopter (see figure 1). The model has the

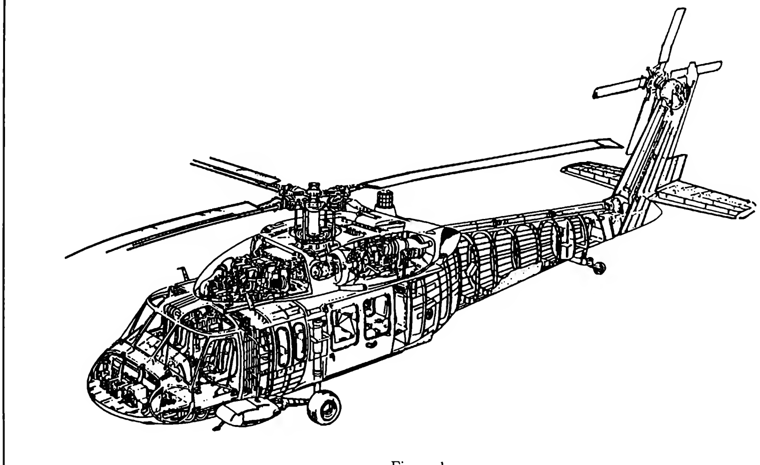


Figure 1.

capability to be extended for pilot-in-the-loop purposes. It features submodels for the main rotor, tail rotor, engine, fuel control system, fuselage, empennage, flight path stabilization (FPS), stability augmentation system (SAS), and mechanical control system. By far the most complex subsystem, the main rotor model features 4 rotor blades that are divided, for computational purposes, into 5 elements per blade (although the model can easily be changed to a different number of blades or a different number of elements per blade). Blade element theory is used to describe in detail what happens at each rotor blade element. Given the rotor controls, the blade flapping and lagging dynamics, the blade dynamic twist, the preformed blade shape, and the rotational speed of the rotor, it is possible to compute the velocity and attitude of each blade element at any point in time. Figure 2 shows velocity and angle of attack of a blade element. Due to rotor rotation and the downward-pointing induced velocity, the blade element "sees" air coming toward it. As a result, the rotational velocity and induced velocity vectors yield a velocity vector that is pointing slightly downward. Note the blade element angle of attack α and the blade element pitch angle Θ. Lift is, by definition, perpendicular to the resultant velocity. The horizontal component of lift is the induced drag. The induced lift and drag forces, multiplied by the distance of the blade element from the rotor hub, contribute to the torques in the rotor shaft. The contributions of all blade elements yield the total induced torques. Repeating this for the lift and drag forces and summing over all the blade elements yields the total forces on the rotor hub. The total forces and moments are inputs to the equations of motion of the airframe. As stated before, the blade element method is computationally intensive: For a rotor with 4 blades and 5 elements per blade, there are 1916 multiplications, 1286 additions and subtractions, 36 divisions, 14 sines and 14 cosines, 70 square roots, 3 exponents, 21 arctangents, 4 nonlinear functions of 1 variable, 60 nonlinear functions of 2 variables, and 55 state variables to be computed for each integration step.

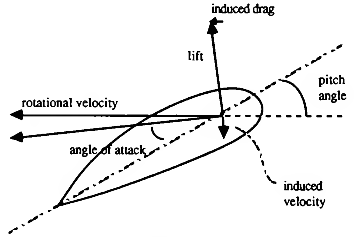


Figure 2.

4 THE SYSTEM 100

The SYSTEM 100 is a powerful computer system designed specifically for the real-time and time-critical simulation of continuous dynamic systems. The SYSTEM 100 consists of hardware and software subsystems. The hardware subsystem consists of the AD 100 computer with a wide variety of optional I/O facilities, together with a general-purpose digital computer of the VAX family. The AD 100 is a 64-bit floating-point multiprocessor, which is particularly well suited for numerical integration. Its high-speed I/O facilities offer numerous data types and formats allowing complex hardware-in-the-loop simulations with a wide range of external hardware. ADSIM software is used to program this combined system. ADSIM provides both a high-level programming language for the AD 100 and an interactive run-time environment. Some of the key features of ADSIM, which allow much faster and more efficient implementation of dynamic models than other general-purpose languages are the use of nonprocedural code, the treatment of both differential and difference equations, the use of subroutines in the form of FUNCTIONs and MODELs, the ease of programming multivariable function tables, and the inherent interactive nature of the programming system.

5 IMPLEMENTATION OF THE MAIN ROTOR ON THE SYSTEM 100

ADSIM allows both procedural and nonprocedural code, and it is important to understand the difference. Procedural code will be executed in the sequence in which it is written. Nonprocedural code allows the programmer to write the model without regard to the order of execution. Nonprocedural code is sorted by the ADSIM compiler to ensure optimum execution and to ensure that variables used on the right-hand side of an equation are first defined on the left-hand side. Therefore, nonprocedural code allows various programmers to work independently on subsets of a model without concern for the execution order of the model as a whole. Once submodels have been verified independently, they are put together in a single source file, and the ADSIM compiler will take care of correctly ordering all the equations.

The combination of the use of nonprocedural code together with difference equations or next-state variables gives a powerful tool to the programmer as shown in the following example. A Fourier prediction method is employed to obtain the flapping and lagging angles and angular velocities of the main rotor blades. The equations for the flapping angle, β_1, and flapping rate, $\dot{\beta}_1$, at time equals nT for blade 1 have the following form:

$$\begin{aligned}
\dot{\beta}_1(nT) &= \ddot{\beta}_1(nT-T)\frac{\sin\Delta\Psi_r}{\Omega} + \dot{\beta}_1(nT-T)\cos\Delta\Psi_r \\
\beta_1(nT) &= \beta_1(nT-T) + \dot{\beta}_1(nT-T)\frac{\sin\Delta\Psi_r}{\Omega} + \\
&\qquad \ddot{\beta}_1(nT-T)\frac{1-\cos\Delta\Psi_r}{\Omega^2}
\end{aligned}$$

where T is the integration step size, Ω is the rotor shaft speed, $\Delta\Psi_r$ is the azimuthal step size, and $\ddot{\beta}_1$ is the flapping acceleration, which follows from the forces and moments acting on the rotor blade. The implementation in ADSIM uses the next-state operator"#" to obtain the previous frame values of $\dot{\beta}_1$ and $\ddot{\beta}_1$ as shown in the following ADSIM code fragment:

```
i_omgmr          = 1.0/omgmr
somgr,comgr      = SIN_COS(d_psir)
radk1            = somgr*i_omgmr
radk2            = (1.0-comgr)*i_omgmr*i_omgmr
brdotmr1         = radk1*old_brddotmr1+
                   comgr*old_brdotmr1
old_brdotmr1#    = brdotmr1
brmr1            = old_brmr1+radk1*old_brdotmr1+
                   radk2*old_brddotmr1
old_brmr1#       = brmr1
brddotmr1        = ...complicated expression...
old_brddotmr1# = brddotmr1
```

In this code fragment, **omgmr** stands for Ω, **d_psir** stands for $\Delta\Psi_r$, **brdotmr1** stands for $\dot{\beta}_1$, etc. The statements with the "#" operator should be read as having the value of the variable on the left-hand side of the equation in the next frame, which is equal to the value of the variable on the right-hand side in the current frame. Also note that due to the nonprocedural nature of this code it does not matter in which order these statements are written. The ADSIM implementation of the UH-60A rotor uses the next-state variable concept quite extensively, which greatly improves the simplicity and readability of the code. Using next-state variables, it is immediately clear which variable contains a value calculated in the previous frame, while the order of the statements is irrelevant. For example, in FORTRAN, it would not have been possible to identify by name alone whether a variable contains a value from the previous frame; therefore, the reader of the code would have to pay special attention to the order of the statements to be able to understand their function.

ADSIM allows very straightforward programming of multivariable function tables. Unlike other general-purpose languages, ADSIM provides support for linear interpolation on multivariant function tables, a technique used extensively in aerospace simulations. ADSIM supports interpolation on functions of up to 7 independent variables. The lift and drag coefficient of each blade element in the UH-60A rotor model are obtained by performing a two-variable interpolation on a function table. The inputs to this table are the blade element angle of attack α and the blade element Mach number M. For the lift coefficient C_L, there are 41 breakpoints for α defined and 11 breakpoints for M. For the drag coefficient C_D, there are 49 breakpoints for α and the same 11 breakpoints for M. In order to perform these interpolations, some declarations must be made in the ADSIM source file:

```
INTERPOLATION_INTERVALS mach_data (11 OF 65)
INTERPOLATION_INTERVALS alfa_1_data (41 OF 65)
INTERPOLATION_INTERVALS alfa_2_data (49 OF 65)
INTERPOLATION_FUNCTIONS lift_coefficient(
                        mach_data,alfa_1_data)
INTERPOLATION_FUNCTIONS drag_coefficient(
                        mach_data,alfa_2_data)
FILES mach_data='MACH.BPT',
      alfa_1_data='ALFA_1.BPT',
      alfa_2_data='ALFA_2.BPT',
      lift_coefficient='LIFT.FUN',
      drag_coefficient='DRAG.FUN'
```

The code fragment above identifies **lift_coefficient** as a two-variable function with 11 breakpoints for the first argument and 41 breakpoints for the second argument. It also states that the numerical values for the 11 breakpoints of Mach number M can be found in a file with the name **MACH.BPT**, the 41 breakpoint values for α can be found in the file **ALFA_1.BPT**, and the 11 × 41 values for the lift coefficient C_L can be found in the file **LIFT.FUN**. Once these declarations are established, the only thing remaining to be done to obtain the values of the lift and drag coefficient of the first element in the first rotor blade is to compute the angle of attack and Mach number of that particular element and then use them as inputs to both two-variable functions:

```
machmr11 = ...complicated expression...
aftfmr11 = ...complicated expression...
afywmr11 = ...complicated expression...
cl11     = lift_coefficient(machmr11,aftfmr11)
cd11     = drag_coefficient(machmr11,afywmr11)
```

Both code fragments above show that multivariant function evaluation in ADSIM is very simple to implement. The AD 100 also executes this code quickly: The 2 interpolations, together with the binary search procedures necessary to find the location of **machmr11**, **aftfmr11**, and **afywmr11** with respect to their breakpoints, take 4.6 microseconds total. The ADSIM implementation of the UH-60A rotor uses this method of evaluating aerodynamic tables extensively. This approach greatly improves the readability and maintainability of the resulting code. In a general-purpose language, it would not have been possible to have access to the functionality shown above; instead, the programmer would have to write the binary search and interpolation routines and also ensure that the appropriate values for the breakpoints and function values would be incorporated in the code.

ADSIM offers an interactive environment for a simulation. As opposed to other general-purpose languages, ADSIM allows you to change every numerical value in the model interactively. This means that without going through a long and tedious recompilation of the model, one can change, for instance, the weight of the aircraft, the function data for the lift coefficient of a blade element, the integration algorithm used for a particular state equation, the integration step size of the model, the length of the run, etc. It is also possible to create graphical output of any variable in the model (see figure 3) or to send any variable to any output channel of the AD 100. These capabilities and the interactive capabilities mentioned above greatly decrease the amount of time required to implement, debug, and fine-tune the simulation.

6 RESULTS FOR THE MAIN ROTOR MODEL SIMULATION

The actual implementation of the main rotor model in ADSIM on the AD 100 is fairly straightforward. While the original FORTRAN implementation of the main rotor model uses 3200 lines of almost undocumented source

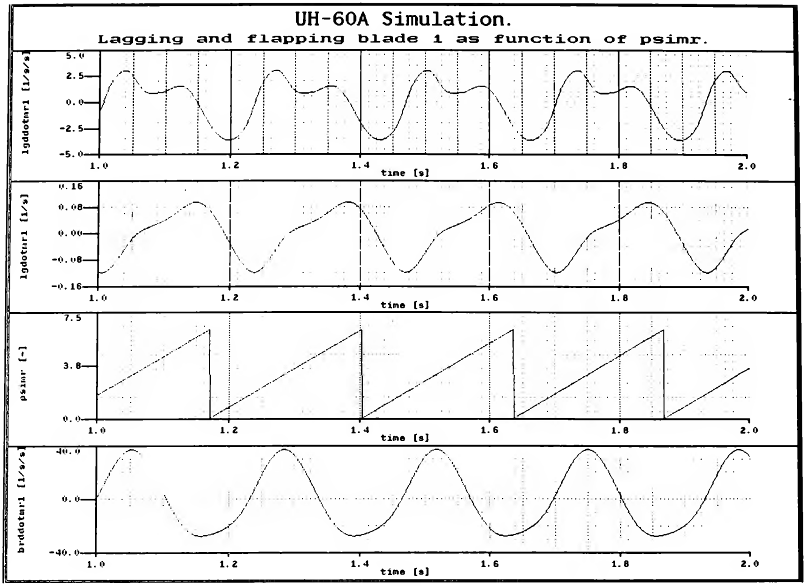


Figure 3.

code, and the average time for an integration step on a VAX 11/780 computer is 79 milliseconds, the ADSIM blade element model of the main rotor fits easily into a single AD 100 and takes only about 2000 lines of highly documented source code. The ADSIM simulation can be solved in real time with an integration step size of 832 microseconds. For a given rotor speed of 27 radians per second, this translates into an azimuthal step size of roughly 1.3 degrees. For reasonable performance and accuracy in a handling-qualities simulation, it has been shown that 4 to 6 elements per blade are sufficient, with an azimuthal step size of 12 degrees.[2] This particular model uses roughly 12.5% of the program memory of the AD 100.

7 RESULTS FOR MORE COMPLEX MAIN ROTOR MODEL SIMULATIONS

The performance numbers for the AD 100 show that it is possible to develop higher fidelity models with more elements per blade and still run in real time. To verify this statement other versions of the Black Hawk main rotor model with 10 and 20 elements per blade were implemented. The main rotor model with 4 blades and 10 elements per blade can run in real time on the AD 100 with an integration step size of 1.34 milliseconds, which corresponds with an azimuthal step size of 2.1 degrees. This model uses 19.4% of the program memory of the AD 100. Increasing the complexity of the main rotor model to 4 blades with 20 elements per blade, gives a model that runs in real time with an integration step size of 1.84 milliseconds, which corresponds with a step in rotor azimuth of just over 2.8 degrees. This model uses roughly 27% of the program memory of the AD 100.

8 RESULTS FOR A FULL-BLOWN HELICOPTER SIMULATION

When we combine models for the UH-60A tail rotor, engine and fuel control system, fuselage, empennage, SAS, FPS, and mechanical control system to the main rotor model with 4 blades and 5 elements per blade, we get a complete helicopter simulation. The FORTRAN implementation of the complete simulation requires over 19000 lines of source code. The average integration step size on a VAX 11/780 computer is 94 milliseconds. On a CDC 7600 computer, the average integration step size is 20 milliseconds. The ADSIM implementation of the complete helicopter simulation fits easily into a single AD 100, and contains 4400 lines of code. Real time can be achieved using an integration step size of 1.29 milliseconds, which corresponds with an azimuthal step size for the main rotor of roughly 2 degrees, which still provides very good accuracy. The full-blown helicopter model uses 19.1% of the program memory of the AD 100. When we combine the Black Hawk rotor model with 4 blades and 20 elements per blade with the models for the tail rotor, engine, fuselage, empennage, and flight control systems, the overall time for an integration step increases to 2.3 milliseconds. This allows the AD 100 to run this model in real time with an azimuthal increment of roughly 3.6 degrees, which is still significantly better than the minimum requirements described by Houck.[2]

9 MULTIPLE HELICOPTERS SIMULTANEOUSLY

Like most languages, ADSIM allows the use of subroutines in the form of FUNCTIONs and MODELs. However, in ADSIM a MODEL is a subroutine that is treated as nonprocedural code, while a FUNCTION is a subroutine that contains procedural code. This subtle difference between ADSIM and other general-purpose languages can have dramatic effects on the code of a simulation model. For example, it is now possible to develop the ADSIM code for the main rotor and put this code in a MODEL. Other building blocks of a helicopter simulation, like the flight control systems, can also be written in MODEL form. Because MODELs are treated as nonprocedural code, correct treatment with respect to computational flow and numerical integration will be taken care of automatically by ADSIM. The overall helicopter model can then be built up from a variety of these building blocks. Then, for simultaneous helicopter simulations, the code for a helicopter can be put in a MODEL again, and this MODEL can be invoked multiple times in the ADSIM simulation to describe the engagement of multiple helicopters. It should be clear that this technique allows for rapid transitions from a helicopter model with 5 elements per blade to a helicopter with 20 elements per blade to 2 helicopters with 20 elements per blade. The following results for multiple helicopter scenarios were obtained. Two helicopters with 5 elements per blade can run in real time on the AD 100 with an integration step size of roughly 2.5 milliseconds. Extrapolating this result, one can predict that it is possible to run 5 of these helicopters simultaneously in real time on a single AD 100 while still maintaining reasonable accuracy for a handling-qualities simulation. Two helicopters with 20 elements per blade can still be simulated simultaneously by the AD 100 with an integration step size of 4.5 milliseconds.

10 SOME I/O ASPECTS

In some cases, the interconnection of a number of high-speed general-purpose computers, for the purpose of real-time helicopter simulation has been implemented. More than one general-purpose computer may provide enough computational horsepower to solve a helicopter model in real time; however, real-time synchronization, timing, and I/O between these computers are newly created and sometimes very serious problems to be overcome. To avoid most of these timing and synchronization problems, the SYSTEM 100 offers and I/O processor that is separate from the other AD 100 processors. This processor, called the Input Output Control Processor (IOCP), works in parallel with the rest of the AD 100, which is busy solving the helicopter model. Under separate program control, the IOCP will send variables to and collect variables from the external world at specified rates. The communication between the AD 100 and the IOCP is via a dual-ported memory. The AD 100 writes variables for output to this dual-ported memory, and the IOCP reads the variables from the dual-ported memory at a specified rate, performing format conversions when necessary and sending the results to the AD 100 I/O system. Once in the I/O system, the results may be converted to analog signals or they may be dispatched to other digital hardware. The IOCP works at the same basic speed as the AD 100. Consequently, it offers high I/O bandwidth. Also note that the AD 100 does not waste time on I/O-related issues. For example, the time required by the AD 100 to read a variable from or write a variable to the IOCP via the dual-ported memory is roughly 200 nanoseconds. Except for this small delay, no other overhead is introduced in the AD 100 program by I/O. Therefore, the total time added to the ADSIM implementation of the UH-60A for I/O is minimal.

In the very near future, the AD 100 will have another separate processor, called the Communications Link Processor (CLP), which will provide Ethernet and FDDI fiber-optic protocols. The CLP will also be able to work in parallel with the rest of the AD 100.

11 CONCLUSIONS

The performance numbers for the AD 100 show that it is possible to use higher fidelity models for the UH-60A with up to 20 elements per blade and still run in real time. Conversely, since the rotor simulation constitutes the major portion of the computational requirements in an overall helicopter simulation, it follows that a single AD 100 can be used to simulate several helicopters simultaneously in real time, as required in simulating the engagement of multiple helicopters in combat. Because of the simulation-oriented features available in the language, ADSIM also allows a very fast transition from a single- to a multiple-helicopter simulation.

12 REFERENCES

1. Howlett, J.J., "UH-60A Black Hawk Engineering Simulation Program", NASA CR-166309, December 1981.

2. Houck, J.A. "Computational Aspects of Real-Time Simulation of a Rotary-Wing Aircraft", Masters Thesis, George Washington University, May 1976.

88-4584-CP

PRESENT AND FUTURE DEVELOPMENTS OF THE NLR MOVING BASE RESEARCH FLIGHT SIMULATOR

C.J. (Hans) Jansen*
National Aerospace Laboratory NLR
Anthony Fokkerweg 2
1059 CM Amsterdam,
The Netherlands

Abstract

The paper presents an overview of the upgrading programme of the NLR Flight Simulator. The avionics system consists of: an ARINC bus interface system to couple o.a. EFIS displays, a general-purpose graphics station, and a programmable EFIS. The new fully hydrostatic 6 degrees-of-freedom motion system with high bandwidth (only 45 deg phase lag at 4 Hz for acceleration commands from the simulator computer) is described in more detail. Finally the digital motion interface and the proposed bus interface system are described.

1. Introduction

For studying a broad field of problems related to pilot-aircraft interactions, NLR operates a versatile moving base research flight simulator at its Amsterdam facility.

Investigations are being performed related to:

* pilot-aircraft integration;
* handling qualities;
* display systems;
* flight simulation techniques;
* advanced flight control systems;
* operational aspects.

The simulator equipment consists of several modules:

- a multi-processor computer system,
- a TV-modelboard visual system,
- a four-degrees-of-freedom motion system,
- a transport type aircraft cockpit, serving a 2-man crew with the possibility of an additional observer seat,
- a single-seat fighter cockpit,
- a control desk,
- recording equipment,

Present and future research projects:

* evaluation of primary flight displays of a civil transport aircraft;
* approach and departure procedures/techniques for MLS equipped runways;
* handling qualities guidelines for the design of future transport aircraft equipped with fly-by-wire systems and Head Up Display;
* handling qualities of high-performance fighters and helicopters.

Demands from these projects triggered an upgrading programme for the flight simulator.

2. Upgrading programme

The current upgrading programme consists of installation of new:

* avionics systems	
- ARINC bus interface system	1985
- installation of standard EFIS instruments	1986
- multi-purpose image generating system	1987
- programmable EFIS	1988
* second motion system with 6 degrees-of-freedom	mid 1988
* interface system	1989

3. Avionics systems

ARINC bus interface system

The ARINC bus interface system (ABIS) enables to actual ARINC 429 aircraft hardware (standard EFIS Primary Flight and NAV displays) to be coupled to the simulator.
ABIS is built around a separate mini-computer, which also performs the data formatting.

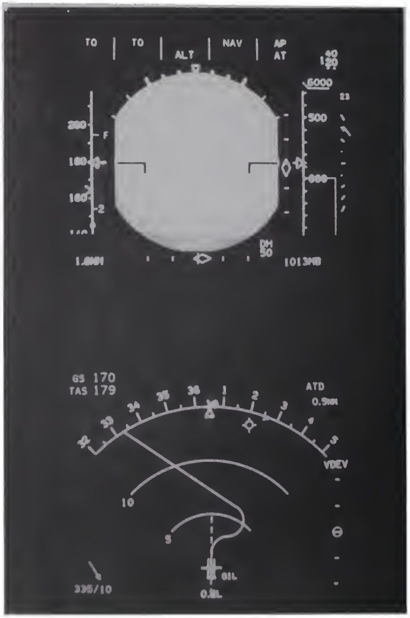


Fig. 1 EFIS displays (PFD+ND) as used during MLS departures

[*] Senior Engineer, Flight Simulation Group

Standard EFIS

Two standard Collins EFIS displays for civil transport aircraft are installed in the cockpit. This EFIS has been used for:

. evaluation of display formats,
. NAV display during simulation of MLS departures (Fig. 1).

Programmable image generation

NLR operates two programmable image generation systems for aerospace research: a general-purpose Silicon Graphics Integrated Raster Imaging System (IRIS) and a Sperry programmable engineering Electronic Flight Instrument System (EFIS).

- IRIS

The general-purpose IRIS is used in the laboratory for fast stand-alone evaluations or coupled to the flight simulator facility.
This system features:

* powerful Motorola MC 68020 processor with 4M byte memory;
* Ethernet local area netwerk for high-speed communications and interfacing with the host computer;
* FORTRAN and C compilers for high-order-languages application development;
* Geometry Pipeline for real-time graphics;
* Standard video display: 19 inch diagonal, 60 Hz refresh rate, non-interlaced;
* alternate video interface according to CCIR standard, externally synchronizeable, permitting video mixing with other sources (e.g. in head-up presentation) and video recording;
* 7 by 5 inch display units available for head-down presentation in the cockpit.

As a thorough evaluation of flight displays demands consideration of the dynamic aspects, NLR is going to extend the interface in the stand-alone situation with an analog joystick adapter. This enables flights with a (simplified) aircraft (+ flight controls) model and dynamic evaluation of the proposed display formats.

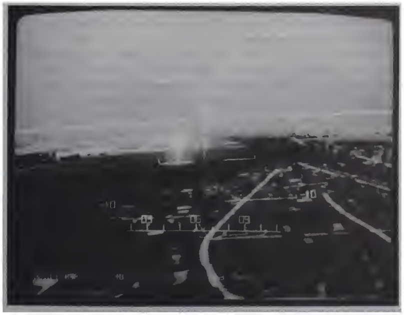


Fig. 2 Example of Handling Qualities project Head Up superimposed on signal from visual system

During a recent project on handling qualities of fly-by-wire aircraft IRIS was used to generate the Head Up Display (see Fig. 2).

- Programmable engineering EFIS

The programmable EFIS consists of two Display Units (DU) with integral Bezel Control Panel (BCP), a Remote Control Panel (RCP) and a Display Electronics Unit (DEU).
An important research capability of this EFIS is that the display format is in-house programmable. The system provides the flexibility to define the complete composition of a dynamic display in terms of: parameters, symbols and characters, symbol location, parameter priorities, colours (Fig. 3).

New application software can be developed by NLR personnel on a Software Development Station after which it is downloaded to the DEU through an IEEE-bus.

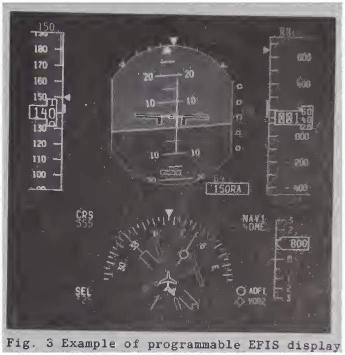


Fig. 3 Example of programmable EFIS display

4. Motion system

The new 6 DOF (degrees of freedom) motion system was designed according to a specification by NLR. Manufacturing is completed and acceptance testing is in its final stage.

Specification

In an initial stage of the upgrading programme the performance requirements of the motion system have been compiled. There have been intensive contacts with operators and users of training and research simulators, and the literature was searched for the latest views with regard to motion simulation.

The purpose of a motion system is to provide to the pilot vestibular motion inputs which influence his behaviour. Such inputs are most important in the assessment of handling qualities and various, pilot-in-the-loop operation procedures of aircraft. The sensation of primary concern is acceleration.

The goal is to reproduce the acceleration in the frequency range of human control, say from 0.05 Hz up to a maximum frequency determined by the dynamic characteristics of the aircraft, with minimal gain reduction and/or phase distortion.

Because of the sensitivity of the human to acceleration any false or spurious accelerations created are highly distracting and may lead to unrepresentative behaviour by the pilot. Stringent requirements for the system are necessary to limit noise and unwanted peak accelerations in all directions, and also to limit accelerations in uncommanded degrees of freedom.

As a start the motion requirements for an FAA Advance Simulation Plan phase III simulator, which allows initial training for new pilots can be regarded as minimum requirements for an engineering research flight simulator. Compared with these requirements the performance of the existing four-degrees-of-freedom motion system is insufficient. Forward and sideway displacement are lacking and the maximum travel in heave direction (2 feet) is too limited.

Of the future research projects on the NLR flight simulator handling quality investigations of high-performance fighters and low-level-flying helicopters will be the most demanding with respect to frequency response and transport delay. If a "standard" motion system is used for helicopter simulation, there is almost no difference with the situation without motion. Only if the bandwidth is increased to about 3 Hz and delays are reduced to 50 ms pilot acceptance and also pilot performance when compared to fixed base performance are greatly improved (Ref. 3).
Modern fighters are highly responsive. The bandwidth of the rigid aircraft motions is determined by the bandwidth of the controls in the cockpit. The upper limit is about 4 Hz as this is the maximum frequency for a pilot to manually move his controls.

A computer simulation of a medium-sized twinjet, flying in turbulence and performing "normal" manoeuvres (from take off via cruise to landing) was used to determine minimum motion system limits. If the accelerations to the motion system have no gain reduction and phase distortion due to washout is limited, a total vertical displacement of at least 2 m is necessary.

The payload to be carried on this system may be any one of a number of simulated aircraft cockpit configurations containing an assembly of crew-seating, controls, instrumentation and visual displays, tailored to the simulation task to be carried out.

The present simulator cockpits are:
- a single seat cockpit of 2000 kg;
- a side-by-side transport cockpit of 2700 kg.

A possible extension in the future might be the placement of two additional display units on either side of the cockpit and placement of interface cabinets at the rear side of the cockpit, which would then total 5000 kg.

The motion system has to be capable of achieving the required performance for payloads ranging between 2000 and 5000 kg.

Overall Performance Limits

Operational Ranges

For each degree of freedom the maximum independent levels of displacement velocity and acceleration, achievable by command of the input signal, without obvious degradation of control due to a proximity of System Limits, must be not less than specified in table 1. (next page)

Acceptance levels for "degraded control" will be based upon the acceleration noise content of sine-shaped motion at amplitudes and frequencies encompassing the Operational Ranges.

Dynamic Performance

This specification for the dynamic performance of the system defines the response criteria pertinent to its use and the acceptance limits for the levels of acceleration noise.

To specify for each degree of freedom the general demand for simultaneous combinations of displacement, velocity and acceleration, the instantaneous values of these three parameters are assumed to be related by sine functions. At a given frequency and with progressively increasing amplitude the maximum of Operational Range for one or more of these parameters will be reached.

For the purpose of specifying and assessing dynamic performance of individual degrees of freedom of the motion system within the Operational Ranges the criteria and procedures of AGARD AR-144 (Ref. 4) are adopted. In particular the Describing Function, Peak Acceleration Noise, Noise Ratio and Dynamic Threshold are defined. They will be measured by an appropriate arrangement of accelerometers. These criteria are relevant to the high fidelity, low frequency performance of the system and are only to apply over the frequency range 0.02 Hz to 4 Hz. They are expressed in terms of the acceleration generated at the centre of rotation of the motion system along or about the defined axis of each individual degree of freedom in response to a specific input signal to that degree of freedom. The input signal is generated in the host computer of the simulator.

The first three criteria are relevant to the steady state response of the system in the frequency domain. In addition a limit is imposed on the acceptable level of interaction between driven and undriven degrees of freedom by specifying maximum levels of "parasitic" acceleration.

Describing Function

The Describing Function H (Ki) is defined as the amplitude and phase relationship between the fundamental of the acceleration output signal and a sine-shaped input signal (See Para 2.3 of AGARD AR-144).

For each of the six degrees of freedom the Describing Function must be such that the amplitude ratio is substantially flat (1 ± 0.1) up to 1 Hz with less than 20° phase lag and lie between 1.0 and 0.7 at 4 Hz with less than 45° phase lag. Between 1 Hz and 4 Hz amplitude ratio and phase lag must lie within the boundaries defined by straight lines joining the above limits.

Peak Acceleration Noise

Peak Acceleration Noise (Ap) is defined as the maximum deviation of the acceleration output signal from the fundamental of the acceleration output signal in response to a sine-shaped input signal. (See Para 2.4 in AGARD AR-144). The Peak Acceleration Noise as measured at 0.5 Hz in any driven axis up to 90% of Operational Range must not exceed 0.2 m/s^2 and 0.1 rad/s^2 for translational and rotational axes respectively.

Parasitic Accelerations

Under the same conditions as stated in the previous paragraph the Peak Accelerations generated in undriven axes must not exceed 0.1 m/s^2 and 0.05 rad/s^2 for translational and rotational movements respectively. (See Para 2.4.3 of AGARD AR-144).

Acceleration Noise Ratio

Acceleration Noise Ratio r_n is defined as: the Standard Deviation of the acceleration output signal from the fundamental of the acceleration output signal, in response to a sine-shaped input signal, expressed as a ratio with respect to the Standard Deviation of the fundamental acceleration output signal. (See Para 2.4 of AGARD AR-144).

In any driven axis at a frequency of 0.5 Hz and over the range of acceleration output from 0.3 m/s^2 and 0.3 rad/s^2, for translational and rotational axes respectively, to 90% of Operational Range, the value of the Acceleration Noise Ratio must not exceed 0.1.

Dynamic Threshold

Dynamic Threshold (Δt) is defined as the time required for the output acceleration to reach 63% of a step input acceleration command. (See Para 2.6 of AGARD AR-144).

At acceleration commands equivalent to 0.1 m/s^2 and 0.05 rad/s^2 or above, for translational and rotation axes respectively, the Dynamic Threshold must not exceed 0.05 seconds.

Positional Accuracy

The static error between actual and commanded platform position must be less than 0.1 percent of full scale.

High Frequency Response

It is required to be able to generate controlled environmental vibration at the payload over the range 4 Hz to 20 Hz via direct outputs of the motion interface. The bandwidth (amplitude ratio 0.7, phase lag 45 deg) shall be 6 Hz.

The specification of the motion system as designed and manufactured by Hydraudyne Systems & Engineering (Boxtel, the Netherlands) is given in table 1. It shows an unique combination of high-frequency response and relatively large vertical motion.

Measurement of the Describing Function of each freedom at a given frequency shall be made at input command amplitude equivalent to 10% of Operational Range for each freedom at that frequency.

Table 1

Specification for the operational limits and frequency response of the 2nd generation hydrostatic motion system at NLR

Operational System limits

	Displacement			Velocity	Acceleration
	pos	neg			
longitudinal	1.72	1.34	(m)	± 0.8 (m/s)	± 8 (m/s²)
lateral	1.39	1.39	(m)	± 0.8 (m/s)	± 8 (m/s²)
vertical	1.01	1.14	(m)	± 0.8 (m/s)	± 10 (m/s²)
roll	30	30	(deg)	± 30 (deg/s)	± 200 (deg/s²)
pitch	29	29	(deg)	± 30 (deg/s)	± 200 (deg/s²)
yaw	41	41	(deg)	± 30 (deg/s)	± 150 (deg/s²)

Frequency response

frequency (Hz)	max. phase shift (deg)	amplitude ratio
≦ 1 Hz	20	1.0 ± 0.1
4 Hz	45	between 0.7 and 1.0

*) with useful load between 2000 and 5000 kg

Hydraulic system

There are three possibilities for the design of the hydraulic servo-actuator (see fig. 4):

- conventional asymmetric
  A big advantage of this construction is a short overall length compared with the useful stroke and a rather simple construction. The disadvantage is the unequality of the piston areas. The difference in area increases when the rod diameter is increased for reasons of improved mechanical stiffness. In most cases a specially manufactured asymmetric servo valve is needed for good control of the actuator.

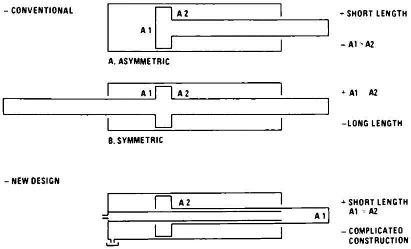


Fig. 4 Comparison of conventional and new-design hydraulic servo-actuators

- conventional symmetric
  The advantage of this actuator is the equality of the piston areas, which also equalizes disturbance forces, improves smoothness of motion and enables the use of "standard" symmetric servo valves. The real disadvantage is the additional free rod end, which almost doubles the overall length. In our existing

four-degrees-of-freedom motion system it was possible, also thanks to the limited travel in heave, to use these kind of actuators to our satisfaction. But when 6 degrees of freedom are needed and the total travel of the actuators is increased it is almost impossible to design a motion system configuration without mechanical interference between all moving parts.

- new design
  This is the so-called double-fold symmetric design with the same, limited, overall dimensions as an asymmetric actuator, but with the possibility of equal areas and associated motion smoothness. The piston areas can be made equal completely independent of rod diameter. This diameter can be optimized for high actuator stiffness.

In order to reduce the friction to an absolute minimum <u>all</u> bearings are fully hydrostatic featuring virtually no metal-to-metal contact. Inside the actuators all piston and rod guides are fully composed of strong fluid films, thus minimizing any static friction (fig. 5). The price for these benefits is a complicated mechanical construction with very small clearance tolerances.

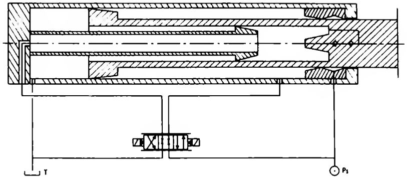


Fig. 5 Fully hydrostatic servo-actuator

Mechanical construction

The motion system is configured as the usual synergistic type (fig. 6), consisting of:
- triangle-shaped baseframe,
- cockpit cradle frame (platform),
- 6 servojacks, positioned as 3 pairs of inverted V's.

The rather large and rigid platform is built up as a boxgirder construction. Finite element analysis showed no resonance frequencies below 20 Hz. Thus, interaction between actuators is low and smooth motion is ensured at high bandwidth. A low centre of gravity reduces the dynamic interaction between motions.

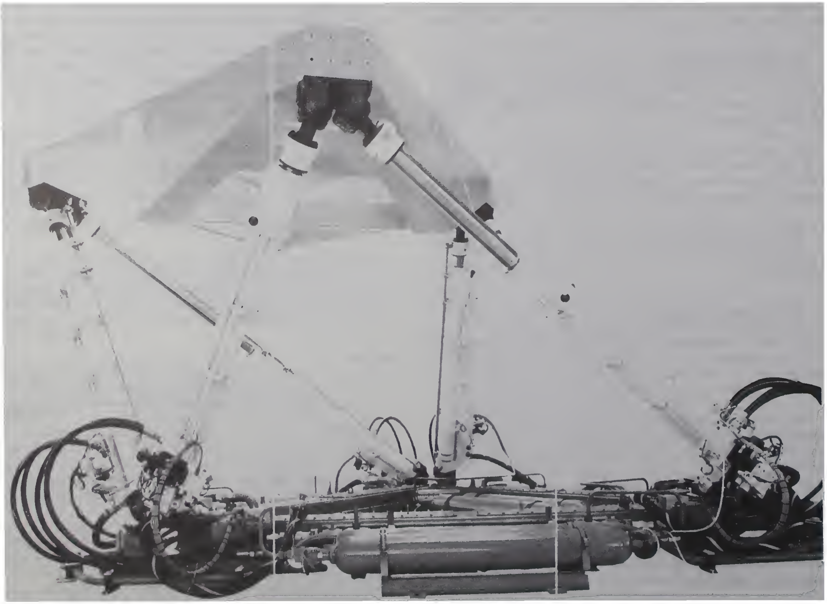


Fig. 6 Mechanical construction of the high bandwidth 6DOF motion system

Both the radial and the axial stiffnesses of the actuators are designed to allow at least 6 Hz system bandwidth. The actuators are attached to the baseframe and platform via almost equal universal play-free cardanic mountings with (enlarged) high rotation possibilities. Very compact dimensions ensure minimal distorsion, good stiffness and a high allowable dynamic load.

Electronic drive- and control Systems

A most important component in the control of a hydraulic actuator is the servovalve. A "critical" valve with slight underlap is a necessaty for the all-important linear behaviour in the zero-flow region of the valve. Lapping to very close tolerances has produced a main valve which generated very small turn-around bumps at flow (= velocity) reversal.

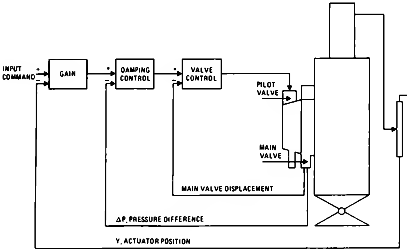


Fig. 7 Block diagram of the servo-actuator control loop

Figure 7 shows the block diagram of the servo-actuator control loop. An ultrasonic transducer is used for position feedback. The gain on the difference between this feedback signal and the input command determines the frequency response. Pressure difference feedback controls the damping of the motion and can be adjusted to compensate for the payload in use. A high damping ratio will suppress the effects introduced by non-linearities and smoothes the motion. As the maximum oilflow through each cylinder is 300 liter per minute the servo valve is composed of a main valve and a pilot valve to improve the control of the actuator at low velocities. The pilot valve has internal electrical feedback of spool position, which improves linearity and bandwidth.

The main valve has a LVDT spool position transducer, which is used in the valve control loop also to improve linearity and bandwidth.
Tests on a separate actuator with a test load of 2000 kg showed a flat frequency response up to 10 Hz.

Motion interface

The MOTION INTERFACE SYSTEM (MOTIF) forms the link between the host computer and the control electronics for the six hydraulic cylinders of the moving base platform. The four main functions of the interface are:

- conversion of the host computer signals (digital words) into analog position and speed signals with a small step size and small step length;
- generation of special effect signals for simulation of engine and propeller effects, e.g.;
- generation of discrete control signals for the hydraulic control unit;
- processing discrete status signals from the hydraulic control unit.

In order to reduce the effect of computational delays of the host computer, accelerations instead of position signals are used as the main drive signals to the motion interface.

The total allowable delay for the asynchronous running MOTIF is 5 ms, this gives a desired iteration frametime of 3.4 ms (the average delay is 1.5 frametime). As the goal is to have equal or less than 45 degrees phase lag at 4 Hz for acceleration signals from the host the response of the motion system itself to the analog output signals of MOTIF should allow a higher frequency, 45 degrees phase lag at 6 Hz is specified.

Each 50 milliseconds the host computer calculates acceleration, speed and position of a specific point on the moving base platform (in general at the position of the head of the pilot). With these data it is possible to calculate the translation of the moving platform itself. Every 50 ms the MOTION SYSTEM receives newly calculated values for acceleration and displacement from the host computer. In order to produce fine cylinder steps the acceleration signal is integrated twice every 3.4 ms into a platform position after which the cylinder extensions are calculated by means of a transformation algorithm. After each data update the calculated platform positions are compared with the received data from the host computer. The difference will be used as a feed-back correction signal in the so-called input "filter" in order to prevent the drifting of the integration process. (fig. 8).

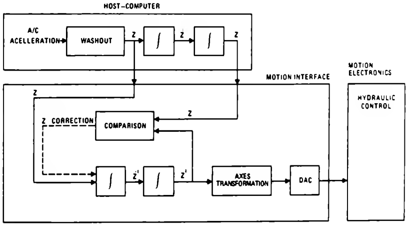


Fig. 8 Motion Interface, functional requirement

As the acceleration signal is available in MOTIF it is possible to use second-order lead filters to improve the total response of MOTIF and motion system.

Information analysis

As stated above the host computer sends platform data to the Motion Interface every 50 ms; this time can be reduced to 10 ms in the future. The data consists of a burst of 16 words with a size of 16 bits each; within this data burst each word has a special meaning. In order to minimize the delay between the accelerations as computed in/by the host and the actual commands to the motion system the total transmission time of this burst must take place within 0.5 ms. The parameters involved with the data burst are:

- 6 accelerations;
- 6 positions;
- special effect buffet frequency and amplitude;
- special effect engine vibration frequency and amplitude.

Apart from the data burst the host computer generates four discrete output signals and receives three discrete input signals for status control of the motion system. Because of the nature of these signals, there is no handshaking involved.

In the start-phase of a simulated flight it is possible to down-load control constants, which replace the default values.

The constants are:

- break frequency and damping of (2nd order) lead filters (adjustable for cockpit mass properties)
- x, y, z coordinates of rotation point (depending on cockpit geometry);
- input filter constants (are depending on host frametime)

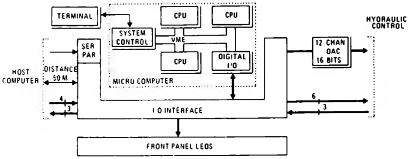


Fig. 9 Motion Interface System

Figure 9 shows the hardware layout of MOTIF. Three CPU's (Motorola 68020) are needed to perform all computations and I/O within the allowable time. The serial connection with the host computer allows the placement of the motion system at a greater distance from the host computer.

A separate logic card in the motion interface checks the frametime of both the data burst as received from the host and the update of the DAC's as output from the microprocessor systems. In good working conditions the so-called SYNC signal is true. If this signal gets false, even momentarily, the independent hydraulic control logic will force the motion system into an emergency stop and cause it to completely disregard all other signals from the motion interface.

The motion interface can also be operated in a local mode, independent of the host computer. Via the terminal inputs are given for degree of freedom, amplitude and frequency of a sinusoide and duration of the test. This allows a stand-alone check on frequency response and noise of the motion system.

Test results

During the Factory Acceptance Test the specified performance (i.e. frequency response, noise and operational limits) has been checked. Preliminary analysis showed that the system meets specifications, except for one point. The frequency responses in x, y and yaw motion are too low. This is mainly due to equivalent loads for certain servo jacks, which are higher than expected and reduce the response.

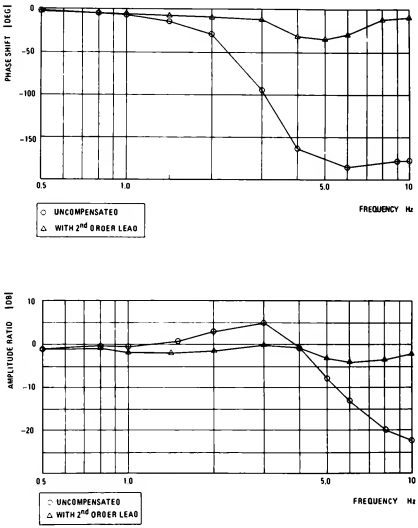


Fig. 10 Frequency response of the 6DOF motion in surge (x)
a. phase shift
b. amplitude ratio

Analysis showed that the performance can be improved by applying a second order lead, which compensates for the measured (approximate) second order dynamics (fig. 10).

Final Acceptance Test measurements will be performed according to AGARD AR-144 (Ref. 5); also the effect of this lead will be checked then.

Figure 11 shows the improvement of the operational limits for heave compared with the 4 DOF motion system. Especially in the lower frequency regions, from 0.05 up to 0.3 Hz, much better motion cues are possible.

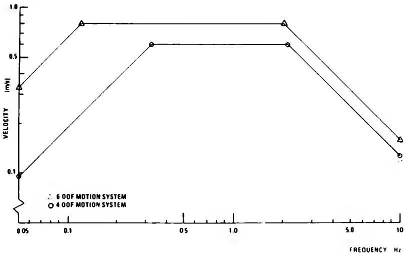


Fig. 11 Operational limits for heave

5. Interface system

Installation of the new motion system (MS6DOF) is the trigger for the development of a new FLIGHT SIMULATOR INTERFACE SYSTEM (FSIS).
FSIS includes not only the interfaces between host computer, motion system, cockpit and visual system but also the definition of man-machine interfaces such as the control desk (CD6DOF) and the sound system.
Apart from research on man-machine behaviour there is a need for research on modern avionics. For these purposes the simulator has to support modern avionics equipment (connection, control, data processing) via standard buses such as ARINC and MIL 1553.

The existing interface converts signals close to the computer and is limited to drive one cockpit at or near the motion system. A new interface system should be capable of driving more cockpits simultaneously (one active, one for preparations). A bus structure with conversion electronics near the cockpits is the most likely outcome of the conceptual phase study (Fig. 12). The motion interface is built up in such a way that only one inputcard has to be modified for adaption to a new interface system.

FSIS will be developed in phases. Phase 1 is the integration of host computer, motion system MS6DOF, transport aircraft cockpit CP2P, new control desk CD6DOF and the sound system. The existing 4 DOF motion system MS4DOF, single-seat fighter cockpit CP1P and existing control desk CD4DOF remain hooked up to the host computer system as they are now (Fig. 13). In later phases they will be incorporated in the new FSIS.

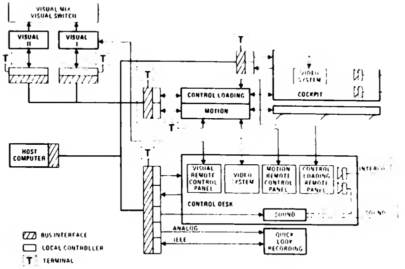


Fig. 12 Block diagram of flight simulator on 6DOF motion

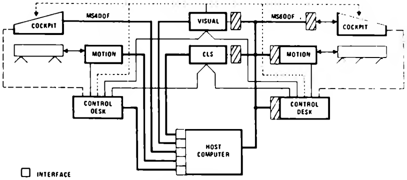


Fig. 13 Integration of simulator on 4DOF motion (present interface) and simulator on 6DOF motion (digital bus-interface)

6. Summary

Based upon our experience with the existing four-degrees-of-freedom motion system a new motion system with improved capabilities (six degrees-of-freedom, total travel in heave 2.1 m) was specified.
Applying fully hydrostatic actuators has resulted in a motion system with high bandwidth and low acceleration noise, which shows good opportunities for demanding tasks like the simulation of high-performance helicopters and fighters.

7. References

1. Erkelens, L.J.J., Flight simulations on MLS-guided interception procedures and curved approach path parameters.
NLR MP 88035 U, Amsterdam, 1988. (Paper presented at the 16th ICAS Congress, Jerusalem, August 1988)

2. Engelen, J.A.J. van, Results of a flight simulator experiment to establish handling quality guide lines for the design of future transport aircraft.
(Paper CP 88-4365 presented at AIAA Atmospheric Flight Mechanics Conference, August 1988)

3. Galloway, R.T., Helicopter training research at the visual technology research simulator. (Paper presented at the RAeS simulator group, 1984 spring convention)

4. AGARD, Dynamic Characteristics of Flight Simulator Motion Systems.
AGARD Advisory Report No. AR-144 (1979)

SPECIFICATION CONSIDERATIONS FOR A SMALL MOTION BASE

88-4587-CP

Robert W. Levi* and Larry Hayashigawa
McFadden Systems, Inc.
Santa Fe Springs, California

Abstract

From a user's viewpoint, specifying performance, loads and other requirements for a simulator motion base holds several pitfalls. Recent experience at McFadden Systems, Inc., with various customer's requirements, provides insight to help avoid some of the difficulties. The discussion will focus upon how to evaluate these characteristics as related to small six degree-of-freedom motion systems.

Several problem areas are associated with the payload. Actual loads on each servo may be much higher than would be expected based solely on payload weight. The "reflected load" is also influenced by the location of the payload center-of-gravity. The causes and effects of this reflected load on hardware structural requirements and performance are presented. A proprietary kinematic and force analysis computer program developed to analyze the effects of reflected loads and center-of-gravity problems is discussed.

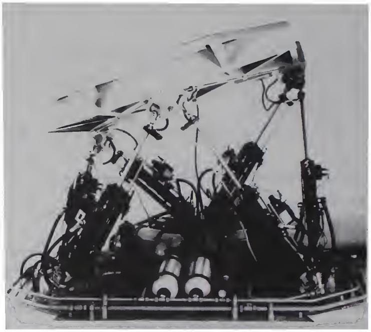
Preventing physical shock, i.e., excessive g's to the payload is an important consideration. An electronic dynamic braking algorithm has been developed to allow each degree of freedom to be used throughout its design range without mechanical limiting. Hydraulic cushion loads during extreme failure conditions are calculated using energy methods.

Introduction

The small six degree-of-freedom motion base is an attractive alternative to a longer stroke equivalent for many applications. See Figure 1. The benefits of a smaller system include reduced facility requirements and lower system cost while maintaining good dynamic performance and payload capacity.

One user who has discovered the benefits of a down-sized system is reported in Reference 1. The rotational axes of the small motion system has performance capabilities similar to those of a larger system, with reduced excursions only in the translational axes.

During the development of McFadden Systems' small hexapod motion base, various techniques emerged for analyzing actuator and structural loading, optimizing and expanding the usable motion envelope, and a braking algorithm simulation. These tools have become very valuable in evaluating customer's requirements and in helping to establish the specification. The following discussion will elaborate on these techniques and the information they provide.



611B MOTION SYSTEM

FIGURE 1

At the outset of a project that requires motion simulation, the motion system specification must be established for procurement purposes and for overall system definition. The specification is usually concerned with required excursions, velocities, accelerations, onset acceleration, payload mass properties, frequency response, etc. The small size of the motion base amplifies the importance of several characteristics that require particular awareness while establishing specifications.

Safety is a prime consideration for all motion systems. Therefore, before mechanical components can be designed, the worst case forces and moments must be known. Two factors that have a major effect on loads are the center-of-gravity height of the payload and the payload leverage or "reflected" load.

Naturally, these factors also have an influence on performance capabilities. Once the user is aware of the importance

*Member, AIAA

of center-of-gravity height and payload weight, steps can be taken during initial phases of the simulator design to minimize these factors.

Center-of-gravity and reflected loads also influence the design of the hydraulic dampers or dashpots that are an integral part of the hydraulic actuators. An analysis technique using energy methods has been developed for calculating forces experienced in the dashpots during worst case failure conditions. These calculations also make use of a proprietary three dimensional vector analysis program that runs on a microcomputer, as mentioned in Reference 2.

For a particular user's load, i.e., center-of-gravity location, weight and inertia, the worst case load conditions can be analyzed to insure that the structural design is adequate. The stress analysis helps guarantee that no parts of the motion system will break in any possible condition. In performing this analysis, the worst case accelerations are also determined. Since the motion platform is essentially a rigid body, the accelerations will also be experienced by the payload. The human occupants on the dynamic platform as well as instruments and optical components must be protected from damaging accelerations. The actuator dashpots are sized to absorb enough energy to limit maximum accelerations to acceptable levels.

Additional protection is available from several electronic circuits. The electronic circuits provide a flexible means for limiting commanded degree-of-freedom acceleration, velocity and position to pre-determined maximum values. Monitoring circuits are available to sense actual actuator velocity and acceleration. A velocity envelope as a function of actuator position can be established to limit the platform momentum and hence the resulting impulse that might occur during an abrupt stop.

Motion System Constraints

Several constraints are inevitable when using a motion system to simulate real time vehicle motions. From strictly a performance viewpoint these constraints are related to motion excursion, motion velocity, motion acceleration, and motion onset of acceleration. Additional constraints are payload load parameters and safety. The relationship of these constraints and their effect on performance and safety will be discussed in the following paragraphs.

Excursion Constraints

Excursion or position is the most obvious of all motion base limitations. "Washout" algorithms are necessary to restrict the platform excursion within the operational stroke of the actuators while relaying usable motion cues to the pilot. The "washout" algorithms create a balance by providing cues, realistic as possible to the pilot without producing anomalous motion cues that can be caused by actuator limiting and/or by the washout algorithms themselves.

In order to define motion excursion it is necessary to reference a common point, called the motion reference point. The motion reference point is the point in space to which all motions are referenced. The location of the motion reference point is at the centroid of the triangular plane formed by the upper universal joints. Motion systems, in actual use, however, do not generally rotate at the motion reference point and the user must consider the motion excursion limitations and simulation impact due to an offset motion rotation point.

Excursion limits are defined by the geometry and usable stroke of the motion system. The excursion capability of a motion base can be altered by changing the upper and lower triangular radii. The increase in excursion of a degree of freedom is usually at the expense of reduction of excursion in another degree of freedom. In addition certain geometric arrangements are unstable and some generate excessively high actuator loads. The ramification of these excessively high actuator loads will be discussed in detail later. The length of the hydraulic end cushions reduces the useable stroke in the actuators, thereby reducing the excursion capability of the motion base. All specifications should state whether or not the excursions include actuator end cushion stroke.

Velocity Constraints

Velocity constraints are dictated by two primary technical factors, they are; safety and the simulation motion performance requirements. Once these two factors are determined the motion system designer can select servo valves, actuators, payload limitations and the hydraulic power unit.

The motion system's velocity capability coupled with the inertial properties and actuator geometry primarily dictates the energy the actuator dashpots will have to absorb during a servo runaway. A motion base with an unusually large payload may dictate a reduced velocity specification and/or redesigned hydraulic cushions in order to keep the energy within the capacity of the cushions. Another alternative is to either rearrange the geometry or use longer stroke actuators. The consequence of overloading the actuator end cushions will be excessive accelerations. These accelerations, if high enough may cause damage to hardware or personnel on the motion base. Excessive energy conditions can be avoided if a careful loads and motion analysis is performed during the early portion of the design process. Specifying large motion velocities without a real performance justification can often drive the cost of a motion system up or make it unsafe. Since velocity limitations during simulation maneuvers will be detected by the pilot, it is extremely important to specify and understand the velocity requirements for good cue generation.

Acceleration Constraints

MIL-STD-1558 provides a criterion for a not-to-exceed acceleration of +,- 2.5 g's. These accelerations may occur when an actuator engages a retract or extend cushion at high velocities. If commanded accelerations are not controlled, hardover commands in the degree of freedom channel can also introduce high acceleration loads into the payload.

The servo actuators are capable of high acceleration when the actuator loads are small. Actuator loads are the sum of the static load and reflected total load. The static loading of the actuators changes only as a function of leg length and geometry. The actuator reflected mass changes not only as a function of leg length and geometry but also as a function of motion from other actuators. A simultaneous motion of several actuators can reduce the actuator reflected load. For an example, consider the McFadden 611B Motion Base with an 8000 lbs payload. With the platform moving vertically at neutral, the actuator inertial loading is only 27 per cent of the total payload mass. The 611B motion base actuator force capacity is 8800 lbs in the extend direction and 4400 lbs in the retract position. In extend, the acceleration capability is:

$Fd = Mr * Aa$ or $Aa = Fd/Mr$
$Fd = Fa - Ws$

where;

Fa = Max actuator force capacity, lbf
Ws = static load on actuator from gravity, lbf
Fd = dynamic force capacity of the actuator, lbf
Mr = reflected mass on actuator, lbm
Aa = actuator acceleration, g's

$Aa = (Fa - Ws)/Mr$

From computer analysis for an 8000 lbs load with the center of gravity located 40 inches above the motion reference point:

Ws = 1799 lbs (static load)
Mr = 2212 lbs (reflected load)

$Aa = (8800 - 1799)/2212$
$Aa = 3.2$ g's

Without actuator acceleration control a hardover command in the heave axis will generate accelerations that exceed the requirements of MIL-STD-1558. It is apparent that with a lighter payload the acceleration levels would be proportionally more. These acceleration levels are generally possible in most motion systems without a method to control or limit accelerations.

On the other hand, single actuator hardovers do not usually generate high load accelerations. This is due to the fact that the single actuator motion reflected loads are very high, therefore, limiting the acceleration capability of the actuators. For example, with the motion system in the neutral position as in the above sample calculation, a single actuator movement results in a reflected load on the order of 5911 lb. The acceleration Aa would then be:

$Aa = (Fa - Ws)/Mr$
$Aa = (8800-1799)/5911$
$Aa = 1.2$ g

The acceleration, Aa of 1.2 g is clearly within structural limits of most modern motion base cabs and well within the requirements of MIL-STD-1558.

Payload/Cab Constraints

The inertial properties of the motion platform payload (i.e., mass, inertia, cross products of inertia) and the center of gravity of the payload are limiting factors in determining the performance and safety requirements of the motion system. Apparent is the fact that performance is reduced when actuator loading becomes large due to static and reflected loads on the actuator. Not so apparent is the fact that the structural safety margins are reduced when the inertial properties and center of gravity distances are increased. Motion system retracted height to center of gravity height and mass contribute to actuator loading. For a given mass and motion base geometry the actuator loading in general increases by the square of the ratio between the center of gravity height (measured from the motion reference point to the payload center of gravity) and the motion base retracted height.

Safety

Paramount in a design or specification of a motion system is the issue of safety. The safety umbrella should not only include the personnel on the motion platform but the safety of platform mounted hardware as well. It is apparent that a structural failure or high accelerations in the actuation system, could result in catastrophic damage to the motion base hardware, visual displays, cockpit equipment, facility, and possible injury to human occupants.

Problem Analysis

Design goals for the small motion base include the ability to use common assemblies to accommodate a wide range of payloads while being able to fit the motion system and its' payload in existing facilities with a ceiling height less than 18 feet.
Early in the development of the small motion system it became apparent, considering the complexity of the six DOF system, that it would be nearly impossible to determine each customer's optimum motion and safety performance requirements without the use of some specialized analysis techniques and some type of computer assisted design tools (programs). In addition, if these tools could be verified through the construction and performance validation of a small motion base, it would give McFadden Systems, Inc. the capability of accurately modeling the motion and safety performance envelope for a motion base of any size or geometry with any type of load.

Type of Analysis

Various analyses performed on the motion system can benefit from computer aided design tools. Some of these are:

Safety - It is important to insure that hardover commands into the dashpot do not exceed its energy capacity. Likewise, maximum velocity capability of the servo actuator, actuator reflected mass and actuator static loading should be adjusted so that dashpot energy capacity is not exceeded.

Motion Performance - The motion system's capability is referenced to the motion reference point. This includes the platform's degree of freedom excursion, velocity, acceleration, and acceleration rate capabilities. Results of the motion performance analysis dictates the actuator, servo valve and hydraulic power supply parameters and servo actuator frequency response requirements. After the actuator reflected loads are determined, then the basic servo actuator frequency response can be predicted.

Clearance analysis - Clearance analysis is the determination of whether the cab/visual display/platform combination clears walls, floor, and/or other obstructions. Additional clearance analysis might include platform mounted domes or other motion base hardware.

Floor loading - This consists of analysis of the force vectors introduced into the concrete on which the motion base is mounted. Significant floor loads can occur when the actuators engage the extend dashpots at high velocity.

Initially McFadden Systems, Inc. relied on a simple degree of freedom to leg length program written in BASIC. This program's algorithm used the conventional closed form "forward transformation" technique used in many motion drive equations. This type of program was adequate to establish the degree of freedom motion capabilities.

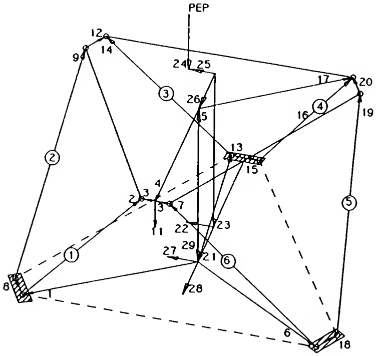
In determining the safety of the system and establishing payload excursion envelopes, it was necessary to do a "reverse transformation", i.e. move the actuators to determine the resultant motion in each degree-of-freedom. The "reverse transformation" and a "mix" of forward and reverse transformation solutions would be very difficult and inefficient to solve using closed form mathematics. Therefore a more generalized computer program for linkage modeling was required. In addition, actuator or leg forces need to be resolved.

Description of MFMAIN (TDKIN)

MFMAIN (McFadden Main) is a three dimensional kinematic and force analysis program developed with personnel at Montana State University. MFMAIN consists of three major sections of code, the preprocessor, a program called TDKIN (Three Dimensional Kinematics) and a post processor. The heart of MFMAIN is TDKIN, which was developed at Montana State University in 1983. TDKIN uses vector loops, dot and cross products to mathematically describe a three dimensional link mechanism, see Figure 2. TDKIN then sets up the equations to solve for position, velocity and acceleration of the links. Since TDKIN was originally developed as a general purpose program it was relatively difficult to use. Consequently, a pre and post processor was developed to make the program "user friendly", to tailor the program for solutions of only non-cascaded six-legged hexapod motion bases and to add some additional analysis features.

The pre-processor first reads an ASCII file created by the user, which contains center of mass, pilot's eye point, actuator end point coordinates and payload mass properties data. After MFMAIN reads the motion data file, the user is then presented with options whether to specify motion in leg lengths or degree of freedom or a mixed combination. The MFMAIN pre-processor then uses the input data for TDKIN for solutions of the unknown vectors. The post-processor then filters out the extraneous vector data and presents the user with only necessary vector data.

The post-processor code also contains leg and motion platform force analysis capability. Access to TDKIN through MFMAIN is still preserved in order for the user to make geometry changes not accommodated by the pre-processor.



VECTOR DIAGRAM HEXAPOD MOTION BASE

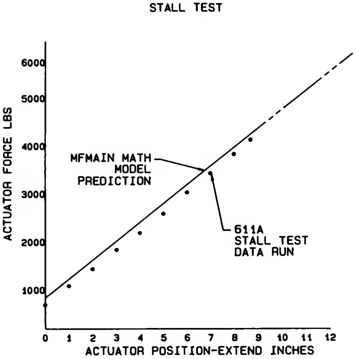
FIGURE 2

Program Validation

In order to gain confidence in TDKIN an informal validation program was developed and implemented. Motion base servo actuators were instrumented with load cells and accelerometers. McFadden Systems' validation of TDKIN consisted of an evaluation of the program's static analysis and dynamic analysis accuracy.

Evaluation of Static Accuracy

Evaluation of static accuracy consisted of two parts. The first part involved the accuracy of the program to predict the leg lengths from a degree of freedom input. The degree of freedom to leg length validation was done first by comparing the results of another motion base motion program which used a closed form solution algorithm unlike the iterative algorithm used in TDKIN. The results agreed. A second degree of freedom to leg length validation was performed on actual hardware, agreement with MFMAIN was within three per cent. The second part of the static accuracy validation was performed on the force analysis section of MFMAIN. The platform was initially positioned at neutral. The actuator under test was retracted and then incrementally extended, while leg positions and forces were recorded. The results of this test and MFMAIN predicted results are shown in Figure 3.



STATIC LOAD VALIDATION CURVE

FIGURE 3

Evaluation of Dynamic Accuracy

Dynamic validation was performed by moving a servo actuator sinusoidally and monitoring acceleration and force. The reflected load was computed by dividing the sinusoidal zero to peak force value by the sinusoidal zero to peak acceleration value. Using TDKIN the leg force was computed by inputting a 1 g acceleration into the appropriate leg. Accuracy was on the order of 80 to 90 percent.

Reflected Mass (Load)

Actuator reflected mass is the apparent inertial loading an actuator(s) senses when acceleration motion occurs. A measure of actuator reflected load, Mr, is the ratio of the actuator force, Fa, to actuator acceleration, Aa. Where the actuator force is the linear force required to achieve a unit acceleration of that actuator or;

$$Mr = Fa/Aa$$

The magnitude of reflected mass on an actuator is a function of leg geometry, leg lengths, mass parameters, and payload center of gravity location.

The implications of reflected load in a linear linkage mechanism such as a six legged motion system is not as obvious nor is the analysis as straight forward as in a flywheel mechanism. The complexity arises because the resultant motion of the load due to actuator motion is difficult to predict. In synergistic type motion bases the reflected load may change an order of magnitude due to platform attitude and motion.

A simplified schematic of actuator reflected load in a single degree-of-freedom linkage system is shown in Figure 4. This figure shows a hinged arm with a weight at a radius of R2 from the hinge point. At a radius of R1 from the hinge point is an actuator. The actuator output rod is tied to the link and the actuator body is tied to the ground. To derive what the reflected load is on the 1 degree of freedom model:

We know that the rotational inertia, Ixx, from a point load, about axis XX is;

$$Ixx = (Mn)\ (R2)^2$$

An effective rotational inertia, Ixx1, of a point mass, Mr, located at the actuator hinge point is;

$$Ixx1 = (Mr)\ (R1)^2$$

Setting Ixx = Ixx1, we have;

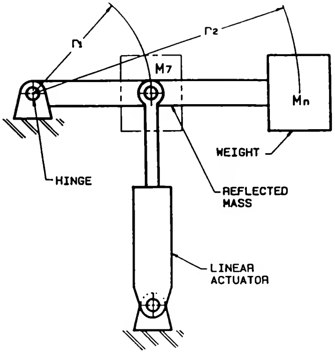
$$(Mn)\ (R2)^2 = (Mr)\ (R1)^2$$

Therefore;

$$Mr = (Mn)\ (R2)^2 / (R1)^2 \text{ or}$$
$$Mr = (Mn)\ (R2/R1)^2$$

Which is the reflected load at the actuator.

It is clear, from above, that the actuator reflected load is sensitive to the ratio of R2 to R1.



SIMPLIFIED DIAGRAM ILLUSTRATING THE LOCATION OF THE REFLECTED MASS IN A THREE BAR LINK.

FIGURE 4

The reflected load characteristic and maximum actuator velocities determine the safety of the system under servo runaway conditions. The energy stored in the moving platform is released when the actuator hydraulic end cushions are engaged. If the platform stored energy is in excess of the end cushion energy capability then high deceleration will occur due to excessive dashpot pressure or when the piston contacts the actuator's hard mechanical limit at excessive velocities. In addition, the reflected load also affects the closed loop frequency response of the servo actuator.

General Observations on Reflected Load

The following general observations are a result of computer studies that were performed for the small motion base. Some of these generalities are also applicable to other sizes of motion bases.

The higher the center of gravity, the higher the reflected loads for a constant payload weight. The increase in reflected load is determined by the geometry of the motion base.

The reflected load on an actuator, for a given platform position, can be estimated by incremental displacement of the platform. The reflected load on an actuator is equal to the total payload (cab and dynamic platform) times the motion of the total payload center of gravity motion squared divided by the actuator displacement squared. For instance, if the geometry of the motion base were altered such that for the same actuator displacement and the payload excursion to actuator displacement ratio increased by a factor of two, the effective or reflected load on that actuator would be four times higher.

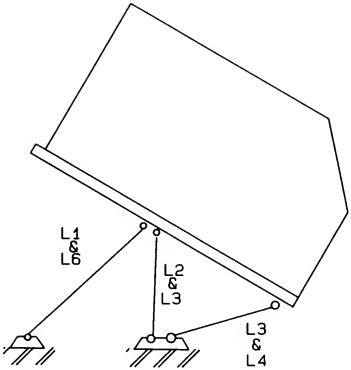
The angle between the floor and actuator, when the motion system is at rest is a relative indication whether the reflected load is high or low. Three different motion base geometries and their reflected loads are shown in Table 1. The "high motion base" configuration geometry is such that the actuator angles are greater than the 611B and likewise the "low motion base" has actuator angles less than the 611B. The results of Table 1 was based on payload weight of 8000 lbs with a center of gravity of 40 inches above the motion reference point.

	Description (Motion Base)	Worst Case Reflected
Config #1	"High"	7,673 lbs.
Config #2	611B	14,474 lbs.
Config #3	"Low"	18,771 lbs.

REFLECTED LOAD VARIATION AS A FUNCTION OF GEOMETRY

TABLE 1

The maximum reflected load occurs when a pair of floor joined legs are extended and the remaining legs are retracted, the greatest reflected loads appears at the two legs opposite the extended pair. See Figure 5. The worse case actuator reflected load can a factor of two greater than the total load in a system with a high center of gravity.



HIGH STATIC LOAD AND HIGH REFLECTED LOAD POSITION

FIGURE 5

Energy Management

Previous experience on various configurations of motion bases indicated that dynamic actuator loading would be an important consideration in a safe motion base design. A careful analysis of the actuation system loading during hardover uncontrolled motion into the cushion stops would have to be conducted before any hardware testing could proceed. Obviously, any loads in the actuation system would also be transmitted to the payload assembly (displays, instruments, trainee). It would be possible to design and manufacture an actuation system that could withstand almost any load. The limiting factor is the payload assembly. As stated earlier, McFadden used the MIL-STD-1558 2.5 g limit as a design goal.

A method was developed to insure that loading into the actuator system and payload did not exceed the 2.5 g limit. During the specification and design phase, various parameters that affect cushions loading are analyzed, they are; 1) maximum velocity capacity of the actuation system, 2) motion base geometry, 3) actuator reflected mass, 4)actuator static load. The assumption was that if the energy capacity of the dashpot was not exceeded, then excessive forces/accelerations would not occur. Assuming that the basic geometry and actuators are fixed parameters, then the only practical methods to control payload energy is to limit actuation velocity and reflected load.

Theory of Energy Management

The following approach was taken to estimate the energy dissipation requirements of the actuator dashpot. This approach was necessary because 1) the reflected loads changed considerably with leg position and 2) it was difficult to estimate the potential energy of the load. This approach was used to compute an adjusted energy capacity of the end cushion(s), which were adjusted on the basis of the static load on the actuator at cushion engagement.

Ea = Total adjusted energy capacity of the cushion

En = Nominal energy capacity of the cushion

Ws = Static load on actuator at cushion engagement length

ld = cushion length

Potential energy of the load is subtracted from En:

$$Ea = En - ((Ws)\ (ld))$$

Since the adjusted energy of the dashpot has to equal the kinetic energy of the load:

$$Ea = (1/2)\ (M)\ (V)^2$$

Therefore, the safe maximum velocity for the given payload is:

$$V = (2)\ (Ea)\ /\ M$$

Application of Energy Management in the 611B Design

Since the actuator and basic geometric layout of the motion base is fixed, McFadden needed to consider actuator terminal velocity and payload mass parameters and how they affect dashpot loading. If payload mass and c of g could not be altered to favorably affect the reflected mass on the actuator then only by reducing the actuator velocity capability could the energy be limited.

For example; If the payload is 6000 lbs on the 611B motion system, then with the motion base in the worst case reflected load position, MFMAIN computes that the reflected load on the actuator is 5000 lbs and the static load is 2000 lbs. If the nominal cushion energy capacity, En, is:

$$En = 16000 \text{ in - lbs}$$

The potential energy of the static load, with a 2 inch long cushion is:

$$\begin{aligned} Ws \times ld &= 2000 \text{ lbs} \times 2 \text{ inches} \\ &= 4000 \text{ in-lbs} \end{aligned}$$

The adjusted energy capacity of the cushion is:

$$\begin{aligned} Ea &= En - (Ws \times ld) \\ Ea &= 16000 - 4000 \\ Ea &= 12000 \text{ in-lbs} \end{aligned}$$

Therefore the safe maximum velocity is:

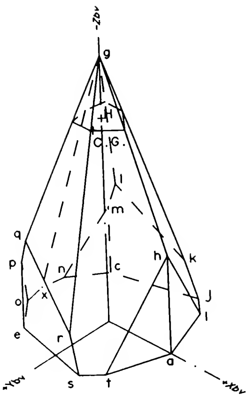
$$\begin{aligned} Vm &= ((2 \times Ea)/M) \\ &= (2 \times 12000/12000/386.4) \\ Vm &= 27.8 \text{ inches/per/sec} \end{aligned}$$

The actuator servo valve or other energy limitation methods must limit the worst case maximum velocity to less than 27 in/per/sec when the piston enters the dashpot to maintain a safe system.

Center of Gravity

Motion systems have practical limitations on the magnitude and location of the payload center of gravity. A center of gravity envelope is determined by the actuator dynamic force margins and static moment requirements. Figure 6 shows a typical center of gravity envelope for a highly loaded motion platform. The center of gravity envelope is referenced to the motion reference point, with the platform fully retracted. The actuator dynamic force margins are set by establishing the allowable minimum system pressure.

Motion bases with hydrostatic bearing actuators are not statically stable under all loading conditions. Unanticipated unloading of actuators can occur from the addition of new simulator hardware and maintenance crew and equipment, which could drastically alter the payload center of gravity. Without hydraulic pressure, hydrostatic bearing actuators can extend under certain load conditions. Since static seals are not utilized in the piston head, leakage can occur between the actuator control ports, permitting the piston rod to move under a net static tensile load. If loading is critical or if interchangeable cabs are to be mounted on the motion platform, then a center of gravity envelope and moment loading information should be computed.



ALLOWABLE C OF G BOUNDARY LIMIT

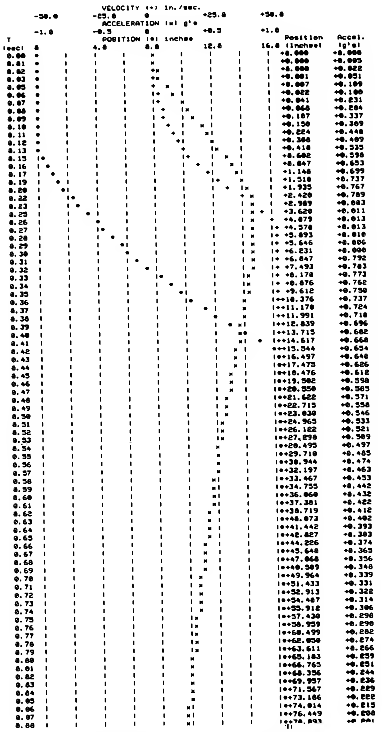
FIGURE 6

Braking Algorithm Simulation

Electronic Braking allows the motion system to be used through its full range of excursion without hitting dashpots or limit switches while smoothly stopping the system as it approaches the stroke limits.

A digital computer simulation was written to assist in evaluation of various braking schemes. The simulation includes the dynamics and non-linearities of McFadden's proprietary algorithm in addition to the dynamics of the servo. The results shown are representative for a degree-of-freedom where all legs move together, such as Heave. This simulation has proven to be quite useful in evaluating braking performance of the system with various washout time constants and servo response.

The following results are shown for a washout time constant of .16 Hz. and a second order servo bandwidth of 2 Hz. Figure 7 represents the response to a 1 g. step acceleration command to the washout algorithm, but without any braking applied. Note that the peak acceleration reached is not quite 1 g., but rather 0.813 g. This is due to the finite response of the servo. Higher response servos will produce peak accelerations that asymptotically approach the command. Granularity in the data is due to the dot matrix printer-plot. The position and velocity curves quickly exceed practical limits of the motion system: over 76 inches of simulated stroke were required with the system still accelerating at .2 g's.

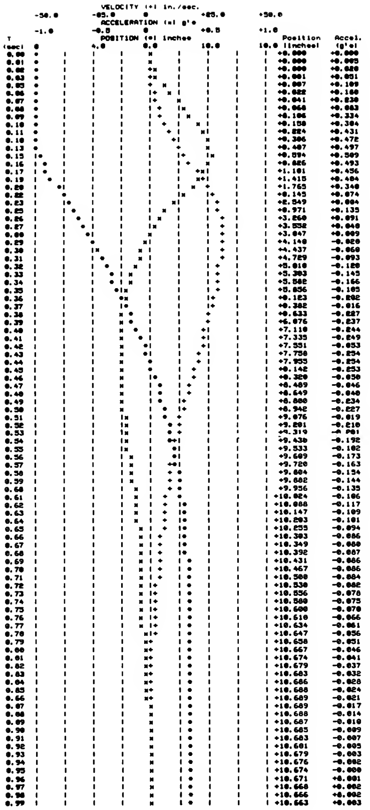


```
  T     Position   Accel.
(sec)   (inches)   (g's)
0.00    +0.000    +0.000
0.01    +0.000    +0.005
0.02    +0.000    +0.022
0.03    +0.001    +0.051
0.05    +0.007    +0.109
0.06    +0.022    +0.180
0.07    +0.041    +0.231
0.08    +0.068    +0.284
0.09    +0.107    +0.337
0.10    +0.150    +0.389
0.11    +0.224    +0.440
0.12    +0.308    +0.489
0.13    +0.418    +0.535
0.15    +0.602    +0.598
0.16    +0.847    +0.653
0.17    +1.148    +0.699
0.19    +1.518    +0.737
0.20    +1.935    +0.767
0.22    +2.420    +0.789
0.23    +2.989    +0.803
0.25    +3.620    +0.811
0.26    +4.079    +0.813
0.27    +4.578    +0.813
0.28    +5.093    +0.810
0.29    +5.646    +0.806
0.30    +6.231    +0.800
0.31    +6.847    +0.792
0.32    +7.493    +0.783
0.33    +8.178    +0.773
0.34    +8.876    +0.762
0.35    +9.612    +0.750
0.36   +10.376    +0.737
0.37   +11.170    +0.724
0.38   +11.991    +0.710
0.39   +12.839    +0.696
0.40   +13.715    +0.682
0.41   +14.617    +0.668
0.42   +15.544    +0.654
0.43   +16.497    +0.640
0.44   +17.475    +0.626
0.45   +18.476    +0.612
0.46   +19.502    +0.598
0.47   +20.550    +0.585
0.48   +21.622    +0.571
0.49   +22.715    +0.558
0.50   +23.830    +0.546
0.51   +24.965    +0.533
0.52   +26.122    +0.521
0.53   +27.298    +0.509
0.54   +28.495    +0.497
0.55   +29.710    +0.485
0.56   +30.944    +0.474
0.57   +32.197    +0.463
0.58   +33.467    +0.453
0.59   +34.755    +0.442
0.60   +36.060    +0.432
0.61   +37.381    +0.422
0.62   +38.719    +0.412
0.63   +40.073    +0.402
0.64   +41.442    +0.393
0.65   +42.827    +0.383
0.66   +44.226    +0.374
0.67   +45.640    +0.365
0.68   +47.068    +0.356
0.69   +48.509    +0.348
0.70   +49.964    +0.339
0.71   +51.433    +0.331
0.72   +52.913    +0.322
0.73   +54.407    +0.314
0.74   +55.912    +0.306
0.75   +57.430    +0.298
0.76   +58.959    +0.290
0.77   +60.499    +0.282
0.78   +62.050    +0.274
0.79   +63.611    +0.266
0.80   +65.183    +0.259
0.81   +66.765    +0.251
0.82   +68.356    +0.244
0.83   +69.957    +0.236
0.84   +71.567    +0.229
0.85   +73.186    +0.222
0.86   +74.814    +0.215
0.87   +76.449    +0.208
0.88   +78.093    +0.201
```

RESPONSE WITHOUT BRAKING

FIGURE 7

The system with the braking algorithm produces a response that is within the capabilities of the motion system, but some trade-offs are required. Figure 8 shows system response with the braking set to stop the system before 11 inches of stroke are used (limits of the 611B motion system). The onset acceleration, or slope of the acceleration curve, are identical with or without braking up



```
  T     Position    Accel.
(sec)   (inches)    (g's)
0.00    +0.000     +0.000
0.01    +0.000     +0.005
0.02    +0.000     +0.020
0.03    +0.001     +0.051
0.05    +0.007     +0.109
0.06    +0.022     +0.160
0.07    +0.041     +0.230
0.08    +0.068     +0.283
0.09    +0.106     +0.334
0.10    +0.158     +0.384
0.11    +0.224     +0.431
0.12    +0.306     +0.472
0.13    +0.407     +0.497
0.15    +0.594     +0.509
0.16    +0.826     +0.493
0.17    +1.101     +0.456
0.19    +1.415     +0.404
0.20    +1.765     +0.340
0.22    +2.145     +0.274
0.23    +2.549     +0.204
0.25    +2.971     +0.135
0.26    +3.260     +0.091
0.27    +3.552     +0.048
0.28    +3.847     +0.009
0.29    +4.140     -0.020
0.30    +4.437     -0.060
0.31    +4.729     -0.093
0.32    +5.018     -0.120
0.33    +5.303     -0.145
0.34    +5.582     -0.166
0.35    +5.856     -0.185
0.36    +6.123     -0.202
0.37    +6.382     -0.216
0.38    +6.633     -0.227
0.39    +6.876     -0.237
0.40    +7.110     -0.244
0.41    +7.335     -0.249
0.42    +7.551     -0.253
0.43    +7.758     -0.254
0.44    +7.955     -0.254
0.45    +8.142     -0.253
0.46    +8.320     -0.250
0.47    +8.489     -0.246
0.48    +8.649     -0.240
0.49    +8.800     -0.234
0.50    +8.942     -0.227
0.51    +9.076     -0.219
0.52    +9.201     -0.210
0.53    +9.319     -0.201
0.54    +9.430     -0.192
0.55    +9.533     -0.182
0.56    +9.629     -0.173
0.57    +9.720     -0.163
0.58    +9.804     -0.154
0.59    +9.882     -0.144
0.60    +9.956     -0.135
0.61    +10.024    -0.126
0.62    +10.088    -0.117
0.63    +10.147    -0.109
0.64    +10.203    -0.101
0.65    +10.255    -0.094
0.66    +10.303    -0.086
0.67    +10.349    -0.080
0.68    +10.392    -0.087
0.69    +10.431    -0.086
0.70    +10.467    -0.086
0.71    +10.500    -0.084
0.72    +10.530    -0.082
0.73    +10.556    -0.078
0.74    +10.580    -0.075
0.75    +10.600    -0.070
0.76    +10.618    -0.066
0.77    +10.634    -0.061
0.78    +10.647    -0.056
0.79    +10.658    -0.051
0.80    +10.667    -0.046
0.81    +10.674    -0.041
0.82    +10.679    -0.037
0.83    +10.683    -0.032
0.84    +10.686    -0.028
0.85    +10.688    -0.024
0.86    +10.689    -0.021
0.87    +10.689    -0.017
0.88    +10.688    -0.014
0.89    +10.687    -0.010
0.90    +10.685    -0.009
0.91    +10.683    -0.007
0.92    +10.681    -0.005
0.93    +10.679    -0.003
0.94    +10.676    -0.002
0.95    +10.674    -0.000
0.96    +10.671    +0.001
0.97    +10.668    +0.002
0.98    +10.666    +0.002
0.99    +10.663    +0.003
```

RESPONSE WITH BRAKING

FIGURE 8

through the first 150 ms. With braking, the reduced acceleration peak of .509 g's is a trade-off that must be accepted in order to limit actuator stroke to reasonable bounds. Note that the amplitude of deceleration pulse is limited by spreading it out over time to produce a smooth stop, with both velocity and acceleration decaying together.

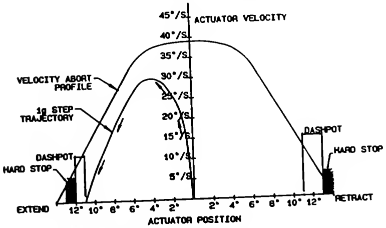
The step acceleration command was used to illustrate the operation of the braking system. More realistic acceleration commands would be represented by sinusoidal pulses, which would allow the braking scheme to produce higher amplitude peak accelerations while keeping within the physical limitations of the motion system.

Velocity Profile Safety Monitor

As discussed earlier, excessive accelerations can occur during failure conditions. The first line of defense against excess acceleration is the scaling of the washout algorithm and the braking scheme. When everything is functional, the combination of the washout and the braking algorithms prevent excess acceleration commands from reaching the leg servos. If an electronic hardware failure prevents this system from working, a second, independent system becomes active: the velocity profile monitor.

The velocity profile monitor works in conjunction with the abort valve. The velocity monitor continuously compares the actuator velocity and position to a pre-programmed profile. If the profile is exceeded, the monitor activates the abort valve. The abort valve hydromechanically forces the actuator to retract at a fixed rate.

Figure 9 shows a plot of a typical velocity profile. This curve is calculated knowing the motion system payload and maximum velocity. If the envelope is exceeded at any point, the abort valve will be able to reduce actuator velocity to a level that allows the dashpot to absorb the excess energy. Plotted within the profile is the trajectory for an actuator responding to a 1 g. acceleration step command as shown in Figure 8.



STEP INPUT ACTUATOR TRAJECTORY

FIGURE 9

Concluding Remarks

Awareness of potential problems with the synergistic hexapod motion system should alert the user to specific areas that require comprehensive analysis to insure the performance and safety of the system. Several areas have been highlighted that may demand a close investigation before a motion system specification is finalized. These items include:

Effects of payload center-of-gravity and reflected mass on actuator loads.

Actuation system acceleration and velocity capacity and the effect on g loading to the payload.

McFadden Systems has developed several analysis techniques for evaluation of possible motion system problems that are available to potential motion system customers when establishing system specifications, including:

A three dimensional kinematic and force analysis computer program.

Energy analysis methods for sizing actuator and cushions.

A program to plot the allowable safe center-of-gravity for the payload.

A computer simulation for evaluating acceleration cues while using washout and braking algorithms.

A velocity profile monitor as backup protection during failure conditions.

Acknowledgments

We would like to thank the Boeing Heli-. copter Company for their patience and assistance during development of the high load capacity 611B motion system. In particular, the technical insights and counsel of Jim Smith and Theo Garnett were invaluable.

Many people, too numerous to mention, at McFadden Systems deserve thanks for their contributions to the 611B development. Bert McFadden and Simon Katiraie have been instrumental in the system design and development.

References

1. A.G. Barnes, "Operating Experience of a Small Six Axis Motion System inside a Dome with a Wide Angle Visual System". AIAA 87-2487, AIAA Flight Simulation Technologies Conference, August 17-19, 1987.

2. Harry W. Townes and Larry Hayashigawa, "Flight Simulator Dynamics", Modeling and Simulation on Microcomputers, 1988. The Society for Computer Simulation International.

3. MIL-STD-1558 Six-Degree of Freedom Motion System for Aircrew Member Training Simulators.

ARTIFICIAL INTELLIGENCE SYSTEMS FOR AIRCRAFT TRAINING: AN EVALUATION

88-4588-CP

Thomas G. Holzman* and Robert W. Patterson**
Lockheed Aeronautical Systems Company
Marietta, Georgia

Abstract

This paper summarizes aircraft operation and maintenance training problems and suggests how they might be remedied through intelligent computer-assisted instruction (ICAI). The paper focuses on a rigorous evaluation methodology designed to facilitate selection of the best artificial intelligence (AI) system for meeting an organization's ICAI needs. A recent application of this methodology revealed that at least a half dozen AI systems possessed the majority of features identified as conducive to good maintenance training. However, no candidate system excelled on all dimensions of value to ICAI. Widespread deficiencies were noted in the areas of interactive videodisc and authoring systems. Trade-offs in features were found between two major classes of computers, known as general purpose workstations and LISP machines.

Introduction

As military aircraft are becoming more sophisticated, they are also becoming too expensive and vital to national security to be set aside for lengthy periods of on-the-job training (OJT). Consequently, aircrew training has come to largely rely on traditional lecture and textbook methods supplemented with exercises on various hardware trainers and high-fidelity flight simulators. Aircraft maintenance instruction also uses traditional lecture and textbook methods, but usually supplements these with instructor-led demonstrations on various side-panel training devices. Serious shortcomings are associated with all of these approaches to training.

While lectures and textbooks can provide valuable information, by themselves they are often too abstract to promote good transfer of training to the job. On the other hand, the superb realism provided by high-fidelity simulators is bought at a high price, resulting in few units being purchased and limited student access.

The laboratory/demonstration experiences used for maintenance instruction have a different set of shortcomings. First, students have difficulty transferring what is learned from the abstract representations of equipment on many side-panel devices to the actual aircraft because of limited physical similarity. Additionally, typical instructor-led group demonstrations make students passive recipients of information. However, research (1) has shown that active student involvement in the instructional process produces the best learning. Similarly, studies of expertise in a variety of contexts, including troubleshooting, have indicated that high levels of ability develop largely through the learner's exploration and extensive practice of new skills. Group demonstrations do little to promote such processes.

In response to some of these problems, computer-assisted instruction (CAI) is gaining increasing acceptance as a supplement to current forms of training. It can provide both aircrew and maintenance students with individualized instruction on basic aircraft concepts and procedures, ensuring mastery of the subject matter through regular testing, immediate corrective feedback, and remediation of deficiencies.

Because CAI is much less expensive per hour of student contact than simulator exercises, it can be made much more accessible. As a result, students can cost-effectively obtain an extensive aircraft orientation through CAI before entering the simulator. The limited simulator time can then be put to its best possible use as a means of practicing complete repertoires of skills in response to sensory cues virtually identical to those of the actual aircraft.

CAI also solves some of the most serious maintenance training problems. By using graphics, digitized photographs, and/or interactive videodisc presentations, the computer can realistically depict aircraft equipment items and their functions, improving the transfer of training from the classroom to the job at a smaller cost than most hardware trainers. Furthermore, because the computer does not present new material until the student answers its queries, active student involvement with the instructional process is ensured.

Development and delivery capabilities for CAI are now being improved through the addition of artificial intelligence (AI) to state-of-the-art instructional computer systems. AI software (e.g., the LISP and PROLOG programming languages, as well as various software development tools) facilitates instructional systems development (ISD) by supporting a technique known as rapid prototyping. This enables designers to assemble innovative programs more quickly than possible with conventional languages (e.g., Pascal or BASIC) and to quickly refine them following successive trials with students. Such an approach greatly facilitates the exploration and evolution of efficient computer-based teaching strategies.

As illustrated in Figure 1, AI also raises the quality of the end-product provided to the student through greater

Principal Investigator, Advanced Training Concepts Independent Research and Development Project

***Associate Investigator, Advanced Training Concepts Independent Research and Development Project*

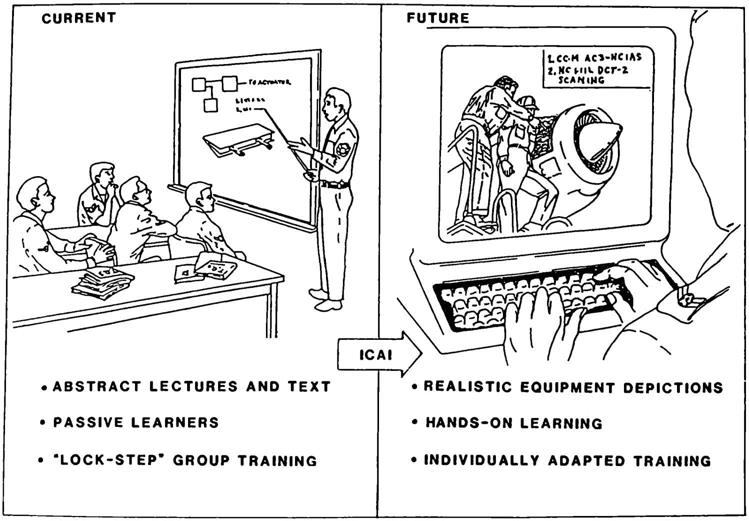

CURRENT
FUTURE
ICAI
• ABSTRACT LECTURES AND TEXT
• PASSIVE LEARNERS
• "LOCK-STEP" GROUP TRAINING
• REALISTIC EQUIPMENT DEPICTIONS
• HANDS-ON LEARNING
• INDIVIDUALLY ADAPTED TRAINING

Figure 1. ICAI remedies aircraft training deficiencies.

realism, expert guidance, and individualization. Known as intelligent computer-assisted instruction (ICAI), this form of training can provide computer-based equipment simulations with remarkable functional fidelity. As suggested by Figure 1's computer screen image of a crew chief guiding a novice technician during OJT, some simulations have the additional advantage of a built-in expert system that can demonstrate for the student the operation or maintenance procedure most appropriate for any particular scenario. Such an expert system also monitors student activities, providing coaching/correction when it detects hesitations or significant deviations from good performance. As such, ICAI performs functions that were once possible only for a few master tutors who were highly knowledgeable of good teaching methods and the subject matter under study.

The System Selection Problem

Given that an organization wishes to use ICAI to meet its aircraft operation/maintenance training needs, an important preliminary problem to be addressed is determining which ICAI system is best suited for that particular organization. Currently, no complete systems are marketed specifically for ICAI. Rather, there is a variety systems capable of performing different AI functions, of which ICAI is only one. Because these products vary widely in both costs and capabilities, they must be systematically evaluated to ensure the maximum return of improved operator/technician performance per dollar and hour invested in training development and delivery.

The Evaluation Process

This paper describes a recommended methodology for rigorously evaluating ICAI systems. As reported by Holzman (2), some aspects of it were first tried and proven by the Lockheed Aeronautical Systems Company to facilitate selection of a conventional (non-AI-based) CAI system for development and delivery of aircraft operation and maintenance training at its Computer Based Customer Training Center. The present paper describes the procedures and findings of a revised process aimed specifically at ICAI. It was tried and proven in the context of selecting an ICAI system for the maintenance training needs of the Company's Advanced Training Concepts Independent Research and Development Project.

Evaluation Team

The evaluation process begins by identifying the individuals best suited to carry it out. Because the sophistication and complexity of ICAI usually requires an interdisciplinary team to develop it (pooling the efforts of computer scientists, educators, graphics specialists, psychologists, and subject matter experts), these same disciplines should be represented among the evaluators to ensure that the system meets the needs of all the development team members. It is in this early phase that the process begins to customize the evaluation to the user organization. For example, maintenance training specialists should participate on the evaluation team if aircraft maintenance will be the ICAI focus, but flight instructors should help evaluate ICAI products for pilot training.

Needs Specification and Candidate Selection

The next phase of the process requires the evaluation team to thoroughly anticipate the ICAI development and delivery activities of their organization and to then specify the kinds of computer features needed to conduct those activities. For example, monitor resolution would be a hardware criterion if simulations with realistic equipment depictions were expected to constitute major portions of the training program. On the other hand, because Common LISP is an AI programming language standard, its availability would be an important software criterion if the evaluation team expected the system to be used extensively for rapid prototyping of instruction.

Usually the specification of system needs interacts with assembly of a slate of candidate systems. By learning about the capabilities of various AI systems, the evaluators begin to consider more features for possible relevance to their particular organization. At the same time, the progressive refinement of anticipated ICAI development and delivery scenarios allows the team to quickly disqualify some systems from detailed evaluation because of glaring deficiencies on criteria deemed essential (e.g., complete implementation of Common LISP, windowing capabilities to facilitate the programming effort, etc.).

Additionally, some systems can be eliminated from serious consideration at an early stage because they clearly exceed budget limitations. However, this criterion must be applied cautiously since the costs of AI systems are falling dramatically and substantial unadvertised discounts are often made available through serious discussions with vendor representatives.

Sources of System Information

As the evaluation team assembles a group of candidate systems that seem capable of meeting their organization's ICAI needs, comprehensive information must be obtained about those products from a variety of sources. Published vendor literature provides considerable technical information. However, product brochures vary in the types of features they describe, requiring the evaluation team to request supplementary information from vendor representatives to assure that comparable information is available for all of the candidates. This is especially true for training-specific offerings (e.g., simulation development software), which are relatively new and often not mentioned in AI marketing literature and presentations.

Very helpful information can also be obtained from current ICAI developers/users with similar objectives to one's own organization. We found excellent contacts of this type within a variety of Government research and development organizations, including the Air Force Human Resources Laboratory, Army Research Institute, National Aeronautics and Space Administration, and Navy Personnel Research and Development Center, as well as within leading universities. Some of these organizations have the added advantage of possessing two or more different ICAI configurations and can thus offer informed, but unbiased, opinions of the relative capabilities of those candidates.

Demonstrations that provide the evaluators hands-on experience are an indispensable source of information about AI systems. Such demonstrations taught us early that many absolute measurements of AI hardware capacities reported in the vendor literature should not be considered in isolation from other properties of the fully operational system. For example, the central processing unit (CPU) word sizes of the candidates we considered ranged from 16 bits to 36 bits, and the clock cycles ranged from 2.67 MHz to 25 MHz, but demonstrations of these systems often showed few detectable differences in the time needed to complete similar applications. This occurred because systems can compensate for relative hardware differences in a variety of ways, including: optimizing the system architecture for LISP processing, taking advantage of highly compact microcode; processing more events per clock cycle; and/or preventing interruptions of running programs through dynamic "garbage collection" procedures to update the contents of virtual memory. Published benchmarks are not good substitutes for acutally seeing demonstrations of applications relevant to the organization's ICAI needs, since different vendors publish different benchmarks and since the benchmarks vary in their relevance to ICAI.

Selection Criteria

A comprehensive evaluation of ICAI systems requires consideration of four major categories of system features. Three of these relate to technical features: hardware, software, and support. The fourth major category is cost. The specific features of importance within these categories must be determined by the evaluation team with respect to the needs of its particular organization (e.g., authoring aids may or may not be chosen as a software criterion according to whether the organization expects to develop its own ICAI, contract its development, or purchase "off-the-shelf" products to meet its training needs). Table 1 lists examples of technical features likely to be of interest to many organizations pursuing ICAI, and Table 2 lists some broadly applicable cost factors. The rationales for including most of the features listed in the tables should be clear, since many of them are also important in conventional computing. However, a few comments will be made on some of these criteria to clarify their special relationships to successful ICAI endeavors.

Hardware. The speeds/capacities of the CPU, main memory, virtual memory, and mass storage are importan in ICAI applications, especially equipment simulations, because the complexity of those applications requires substantial system resources. Should the system be slow, students will be frustrated as they wait for the computer to provide new facts, examples, exercises, or responses to their inputs, since all of these activities must be performed in real time.

Monitor resolution and size also influence instructional quality, since they affect the clarity of the graphics used to depict the aircraft equipment. Color is a valuable, but perhaps less important, feature. It is particularly helpful for

Table 1. Technical Considerations for ICAI Systems

HARDWARE	SOFTWARE	SUPPORT
Central Processing Unit	Standard AI Environment	Warranties
Speed	User Interface	Maintenance
Concurrent Processing	Power & Flexibility	Hardware
Main (Random Access) Memory	Windowing	Software
Virtual Memory	Productivity Time Line	User Guidance
Mass Storage	Optional Software Tools	Training
Hard Disk	ICAI Authoring Systems	Documentation
Floppy	Writing/Publishing Aids	
Tape	Programming Languages	Delivery Schedule
Expansion Slots	LISP	Vendor
	PROLOG	Reputation
Monitor	C	Stability
Resolution	Other	AI Commitment
Size		
Color	Integration of Languages/Tools	
Scanner (Digitizer)	Portability	
Printer	Videodisc Control	
Ports	Graphics	
Office Compatibility	End Product Quality	
Networking	Development Ease	
Reliability	Reliability	
Expected AI Advances	Expected AI Advances	

Table 2. Cost Considerations for ICAI Systems

Hardware
Initial Acquisition
Initial Maintenance
Long-term Acquisition
Long-term Maintenance
Software
Initial Acquisition
Initial Maintenance
Long-term Acquisition
Long-term Maintenance
Training Courses

highlighting, and thus drawing student attention to, a specific portion of a complex diagram or equipment image that is the current focus of instruction.

A scanner, or digitizer, can save substantial graphics creation time for an organization that already possesses a large collection of high-quality aircraft drawings or pictures. In a matter of seconds, these images can be captured into the computer's memory to be presented on the monitor either under program control or upon request by the student during a tutorial or simulation. Although new graphics will need to be created to fully take advantage of some instructionally useful system features (e.g., color highlighting of certain equipment components demonstrated during tutorial instruction), existing drawings and pictures can probably fulfill a significant percentage of ICAI requirements.

A final hardware aspect listed in Table 1 deserving special note is expected AI advances. This refers to upgrades likely to come from the vendor that would effectively prolong the useful life of its AI products. For example, certain AI hardware vendors are developing "LISP chips" that could be inserted into their existing AI workstations and thus accelerate their processing speed. Alternatively, these chip sets could be placed in microcomputers to inexpensively enable them to deliver ICAI developed on dedicated AI workstations, thus rapidly expanding the installed base to which state-of-the-art instruction can be targeted.

Software. The AI environment of a system will influence the efficiency with which ICAI applications can be developed. For example, the user interfaces of some systems cleverly incorporate icons to help the user remember the purposes of various software tools that are available. Likewise, the nature of the tools provided and their mode of access can provide novice and experienced developers alike with powerful and flexible programming support. The presence of debuggers, on-line glossaries of terms, and pull-down menus for selecting utilities without exiting the current process promote efficient application development. Similarly, systems that upon reentry restore the contents of the computer screen to the state it was in before logout can ease the transition between development sessions by reducing what the developer must remember about the previous session and allowing faster resumption of program creation.

The second major software criterion in Table 1, windowing, is related to the AI environment, but refers specifically to the system's ability to display two or more processes at once in different regions of the computer screen. This can add significant convenience to the programming effort, making various system utilities/tools (e.g., file directories, system prompts, etc.) immediately available to the developer. Windowing also improves the quality of instructional delivery by allowing such techniques as using one window to present basic facts about the system under study, another window to provide corrective feedback to the student, and still another window to present a menu of topics/exercises for selection by the student. AI systems not only vary in whether or not they offer windowing, but also in the speed and ease with which the windows can be manipulated (opened, closed, sized, and repositioned), either manually or under program control.

The third major software criterion, productivity time line, refers to the number of months of system familiarization required before the development team can begin programming of useful ICAI materials. The evaluation team must consider the entering skills of likely developers when applying this criterion. All developers require an orientation period to become productive on a new AI system, since development environments (e.g., mechanisms for accessing and saving files, printing screen images, etc.) vary considerably across AI systems. However, this period is particularly long for novice developers, usually entailing six months to a year or more of intense study before productive ICAI programming can begin. The slope of this learning curve varies significantly with the types of software available to support the development effort. For example, user interfaces employing icons that clearly symbolize computer processing options reduce the memorization and reference book use needed to make skilled use of the system. Likewise, on-line help that teaches the use of functions when they are needed facilitates timely use of the system.

The next software feature of interest in Table 1, optional software tools, influences the time required to first become productive with the system as well as influencing continued ICAI development efficiency. Sophisticated, menu-driven, software development tools, known as authoring systems, are becoming available as options for some AI systems. These products support creation of major ICAI segments (e.g., training simulations) without requiring the developer to be familiar with a programming language. Software of this type greatly facilitates rapid prototyping and subsequent testing of training materials. Many systems also offer tools for word processing and even sophisticated desktop publishing, which can be executed from a window on the same screen as an ICAI application. Such features can be quite helpful for writing reports that document ICAI development efforts and their outcomes.

The array of available programming languages should also be considered in evaluating systems for ICAI. For purposes of rapid prototyping, at least one standard AI language (LISP or PROLOG) should be present. However, AI languages typically execute slowly, so a language that permits faster execution (e.g., C) is also beneficial for subsequent translation of the application once a prototype has proven satisfactory and ready for large scale delivery.

The next software criterion in Table 1 is the extent to which the system integrates use of available languages with each other and with software development utilities/tools. Different languages perform certain functions better than others (e.g., number crunching versus rapid prototyping), and it is possible to have the best of all worlds if the languages are implemented to allow procedures written in one language to be called from those written in another, with parameters passing between them. Similarly, certain software development utilities/tools, especially authoring systems, can assist development of an important piece of a total training program (e.g., equipment simulation). However, other parts of the program (e.g., intelligent tutorial introduction to aircraft components, locations, and functions) may lack such aids to support their development and must be programmed in an AI language. Ideally, the software should allow these different parts of the program to be integrated, so that, for instance, simulations created with an authoring sytem could be called from an intelligent tutorial written in LSIP.

Besides the issue of compatibility of different software languages, tools, and utilities with each other, a concern, especially for eventual dissemination of ICAI products, is software portability. By this is meant the ease with which an application can be re-hosted on hardware different from that on which it was originally developed. Portability can be achieved in large part by choosing a system that implements AI software "standards." For example, use of the Common LISP dialect of LISP will enable an application to be more easily re-hosted than if it were written in another LISP dialect. The C language offers a similar advantage. Standards are also emerging for graphics (e.g., "X windows") and for operating systems (e.g., UNIX), and these also deserve consideration.

A final software item in Table 1 that may require some explanation is videodisc control. By providing access to selected audio-visual segments on a videodisc under program control, both tutorials and simulations can realistically present sights and sounds of aircraft equipment that serve as cues for operator/technician actions. In this way, transfer of training from the classroom to the job is noticeably improved. In many cases, videodisc can also reduce development time, since video segments can often be produced more quickly than equivalently complex graphic depictions, especially ones showing moving equipment.

Support. Most of the support criteria in Table 1 require no explanation. Only two features with special relevance to ICAI will be discussed here.

First, user guidance is an important consideration since development of AI applications is a new and complex process that requires substantial time to learn. Additionally, techniques specific to the development of training applications are usually not well documented in user manuals and are seldom covered in standard AI programming courses. Thus, the availability and quality of telephone support from the vendor can be especially important to consider as a source of individual guidance to the ICAI developer. Additionally, informal contacts must be relied upon in many cases to find out how best to utilize the capabilities of a particular AI system specifically for ICAI. To the extent to which a system is already in use in one's own organization or in the instructional research and development community, it has the advantage of offering an installed user network that can mutually support the productivity of all its members, particularly the new ones.

Finally, vendor AI commitment will largely determine the future availability of system upgrades, support services, and an installed base to which to target one's ICAI products. The degree of commitment to AI can be determined through several means: observing the extent to which the respective vendors invest in AI product lines; studying what is written about them in AI trade publications; observing how prominent they are at AI trade shows; and finding out about their AI plans through nondisclosure meetings that some vendors are willing to arrange for prospective customers.

Cost. Table 2 displays cost factors that should be considered when evaluating candidate systems. Expenses should be estimated for all products and services likely to be purchased for each system throughout its life cycle. Although some systems may be relatively inexpensive to purchase because of large discounts, their maintenance agreements, which are usually a percentage of the retail price, may be costly.

If an organization anticipates a growth in its ICAI system during future years, its evaluation team should recognize that expansion costs cannot always be linearly extrapolated from those of the initial acquisition. In fact, systems differ considerably in the cost of expansion. Some products have excellent networking capabilities that enable CPUs or terminals to be added without requiring proportional increases in mass storage, printers, scanners, etc., which can be shared through the network. Similarly, while some vendors sell seat licenses for software, which require payments for each added keyboard and monitor, others provide CPU licenses or site licenses, which allow greater access to the software at little or no additional software expense.

Construction of Evaluation Instruments

To ensure that all candidates are systematically evaluated on all criteria chosen by the evaluation team and to support and document this process, a series of evaluation instruments should be constructed by the evaluation team and tailored to the special ICAI needs and resources of its organization. The nature of these instruments and how to construct them will be described next.

Technical Information Matrix. To assist the rating process, a technical information matrix (TIM) should be developed that lists the evaluation criteria down its left side and the candidate products across its top. An example of a TIM is provided in Table 3 for some fictitious products and an abbreviated list of criteria. The cells formed by the intersection of the candidates and criteria are filled with data that will guide later assignment of numerical ratings to the products and ensure ready access to the same set of information by all evaluators.

Table 3. Abbreviated Example of a Technical Information Matrix

	CANDIDATE SYSTEM		
CRITERIA	AI1	SMART	Super 60
Hardware			
CPU Word	32 bit	16 bit	32 bit
Main Memory	8 mb	4 mb	12 mb
Hard Disk	190 mb	190 mb	80 mb
Monitor	Color 19"	Color 19"	B&W 15"
Software			
Common Lisp	Yes	Yes	Yes
PROLOG	No	Yes	Yes
C	Yes	No	No
Authoring System	No	Yes	No
Support			
Training	Extensive	Extensive	None

Technical Evaluation Matrix. The next step toward rating the candidates involves construction of a technical evaluation matrix (TEM). Like the TIM, this matrix lists criteria down its left side and candidates across its top. However, instead of filling its cells with descriptions of product capacities/qualities, the evaluators will use the cells to record assignments of numerical ratings of the candidates with respect to the various criteria.

The evaluation team must differentiate the extent to which each system characteristic will likely contribute to the success of the ICAI endeavor, so that systems excelling primarily in minor features will not be rated as high as those with strengths in more powerful features. Consistent with limitations on human information processing (3), the evaluation team should assign a number from 1 (for least important) to 7 (for most important) for each criterion. The expertise and special perspectives of each evaluator can be incorporated into the weighting process by operating in a "nominal group" mode (4). This requires that evaluators participate in a group discussion of the relative importance of each criterion, immediately followed by individual assignment of a weight to the criterion under discussion. To prevent undue influence by any one individual during the weighting process, group members should take turns initiating the discussion of weights as they work their way through the list of criteria (2). These weight assignments are then averaged across evaluators for each criterion, and the mean weight is rounded to the nearest integer and placed next to its respective criterion in the TEM.

Cost Information Matrix. Cost criteria for the systems, such as those listed in Table 2, should also be listed in a matrix, analogous to the TIM. However, this cost information matrix (CIM) will have known and projected costs in its cells rather than information on technical capabilities.

Cost Evaluation Matrix. As was done for technical characteristics in the TEM, cost factors should be weighted for relative importance. In most cases, near-term costs, which are better known, will be weighted more than future costs (e.g., for expansion), which are less well defined. Likewise, more expensive items (e.g., hardware acquisition) should receive more weight than those likely to require a relatively small budget (e.g., LISP training classes for new ICAI developers). Once determined, these weights are placed next to their respective cost factors to complete the cost evaluation matrix (CEM).

Rating Process

Rating Assignments. The evaluation team functions as a nominal group to actually rate the products, as it had done earlier in weighting the factors in the evaluation matrices. With each member possessing the same set of information and evaluation matrices, a round robin discussion of the merits of each candidate on each criterion proceeds, followed by individual assignments of a numerical rating to the candidate just discussed with respect to the criterion just discussed. This numerical rating can range from 1 to as high as the previously established weight appearing next to the criterion on the evaluation matrix. As described by Holzman (2), to minimize product bias (e.g., halo effects), ratings should be assigned to all products for a particular criterion rather than rating one product at a time across all criteria. Additionally, the order of product consideration should systematically change with each succeeding criterion, so that no product suffers or benefits from always being considered first, last, or in the middle.

Rating Aggregation. Following the rating of all products on all criteria, the numbers in each cell are averaged across evaluators and placed in the corresponding cells of a team-wide TEM and CEM. Next the team-wide TEM and CEM ratings must be aggregated across cells to yield overall scores for each product.

There are several ways to aggregate across criteria. One is to simply compute a grand total or a grand mean of ratings across all technical and cost criteria. Another viable method is to compute totals or means separately for the TEM and the CEM, basing the system recommendation primarily on technical merits and using cost to influence system selection only when the TEM ratings of leading candidates are very close or when a candidate exceeds current or projected budgets. However, a potential problem with either of these two procedures is that the final score is most influenced by system categories (e.g., hardware) that have been subdivided into the most features and thus contribute the most ratings to the total or the mean. It is possible that the evaluation team deems another category to be of equal or greater importance to the overall success of the ICAI endeavor, but it may simply have fewer discernible criteria. Another shortcoming with respect to simple totals is that they do not indicate how close a product is to the maximum score possible, so it isn't immediately clear how well a product fulfills the organization's needs, regardless of its relative standing (perhaps all products are marginal). On the other hand, means have the problem of clustering close together, due to the restricted 7-point rating range, making most product score differences look rather insignificant.

Perhaps the best way to aggregate is to convert candidate scores to a 100-point scale, specifying in advance the relative contribution of each criterion category to the total. For example, the evaluation team may decide that hardware and software are of equal importance and that they are twice as important as support and cost. Consequently, hardware and software would each contribute up to 40 points for the grand total for each product, and support and cost would contribute up to 20 points each. As described by Holzman (2), the scaled rating achieved by a product is computed by dividing the points achieved by the product for a given category by the total points possible for that category, resulting in a proportion that is multiplied by the total points out of 100 allocated to that category; the procedure is repeated for each category, and the category scores are then summed. The result is a number on a scale wide enough to allow perceptible differences to occur between products and that indicates the degree of conformity to evaluation team desires (a score of 100 being the standard for the ideal system).

Results and Conclusions

Scope of the Evaluation

The application of the methodology described here to our own organization's ICAI needs began with an examination of 10 candidates. Four were eliminated early, due to obvious inabilities to fulfill our particular ICAI requirements. The remaining six candidates were rigorously evaluated according to the methodology described in this paper. The hardware for these six systems was made by four prominent computer manufacturers. Two different hardware configurations from the same manufacturer accounted for two candidates, and two different software configurations (one having a third party product) that ran on identical hardware accounted for two more candidates.

The evaluation methodology worked smoothly, although its rigorous fact-gathering and analysis entailed considerable time. A good return on this investment of time was obtained, however, by selection of a system that was relatively inexpensive to acquire and that will save substantial ICAI development time relative to the other candidates.

Conformity to ICAI Needs

As a result of this study, several general facts emerged about the state of AI systems as they relate to ICAI applications. First, all six of our candidates obtained the majority of the 100 points points that could be awarded, but even the best ones were lacking on a substantial number of criteria. Thus, several systems on the market can meet many ICAI requirements, but none come close to being ideal for this application, at least as it relates to the aircraft operation and maintenance training needs of our organization. The nature of these current AI system strengths and weaknesses for ICAI, as well as expected near-term developments, will be described next.

Interactive Videodisc. The most widespread deficiency we observed related to interactive videodisc. As pointed out earlier, many conventional CAI systems use this medium to realistically depict equipment and thus improve transfer of training from the classroom to the job. None of our ICAI candidates offered this capability as a supported product. Fortunately, however, our contacts with AI vendors indicate that this situation may soon be changing. Two vendors had interactive videodisc access in laboratory development at the time of our evaluation, and one of them was willing to provide us with an unsupported version of that software. The other vendor plans to bring out fully supported interactive videodisc software during 1988 as an upgrade to its currently available training simulation authoring system.

Authoring System. The training simulation authoring system just mentioned was the only ICAI authoring system offered by any of our candidates, revealing another general deficiency in this new technology. Like interactive videodisc, menu-driven authoring systems are common for conventional CAI development.

The one ICAI authoring system that we found greatly facilitates the development of computer-based training simulations. Through a series of user-friendly, menu-driven editors, developers graphically depict equipment components, describe their rules of operation, and then assemble the graphic objects on the screen of the computer to form a complete system, which students can then execute as real-time simulations. Developers define both normal and faulted modes of behavior for each component, allowing instructors to insert faults in the system that students must then troubleshoot. This simulation software package also includes a generic expert system that monitors student tests and hypotheses, coaching students when they persist in unproductive solution paths. Unfortunately, as good as this authoring system is for producing simulations, neither it nor the software of any other candidate assists development of intelligent tutorial instruction, leaving a major gap in ICAI software that does not exist for conventional CAI.

General System Tradeoffs. With the exception of interactive videodisc and authoring system software, every aspect of the ideal ICAI system could be found among our candidates. However, no single candidate fully possessed all of these desired features. Furthermore, while the scaled ratings obtained for our products were quite close to each other, those numerical similarities resulted from quite different profiles of system strengths. Many of these differences in system profiles were related to the two broad computer classes from which our candidates were drawn.

The first class, general purpose workstations, are super microcomputers or minicomputers designed for use with a variety of scientific and engineering applications. These applications frequently include AI, computer-aided design and manufacturing, and graphics art, the specific capabilities being determined by the particular set of software (frequently third party) purchased with the system. Three of our candidates belonged to this class of computers. The second class of computers, LISP machines, are designed specifically for AI-related applications. They are optimized for AI processing though hardware (e.g., processors designed especially to expedite "garbage collection" functions required for LISP), software (e.g., use of compact microcode to speed execution of LISP functions, provision of software development tools to support LISP programming), or both. Three of our candidates belonged to this class.

The differences in computing approach taken by the two general classes produce some characteristic distinctions between their respective AI workstations. Although there are exceptions, the general purpose workstations tend to excel in hardware capabilities (e.g., memory and mass storage capacity, number of expansion slots, and networking), largely because they must have a very powerful and flexible baseline configuration in order to handle any of a variety of software packages they might need to host. Likewise, the frequent use of these systems for sophisticated graphics display requires that they have a high resolution monitor (frequently color), large memory and mass storage capacities, and excellent processing speeds. On the other hand, because they are customized for AI, LISP machines typically excel in the software they provide for that application. Their AI environments are very conducive to LISP programming, anticipating most forms of support needed by the developer, resulting in a very high level of program development efficiency for AI applications. Similarly, the specialized purpose of these machines results in highly optimized user interfaces for AI, which reduce the time needed by novices to become productive with the system, as well as aiding experienced developers.

Costs

System costs varied greatly among our candidate ICAI systems. Several systems had major price reductions during the six-month period of the evaluation, although the relative standings of the systems remained about the same. With increasing AI capabilities becoming available on high-end personal computers, good-quality AI features should soon be within financial reach of almost any organization desiring them.

Summary

ICAI holds considerable promise for cost-effectively meeting aircraft operation and maintenance training needs that are inadequately addressed by current instructional methods. However, the wide variability in costs and features of AI systems requires that a systematic evaluation methodology be applied to select the system most suitable for a particular organization's ICAI needs and budget. This paper described such a methodology.

The importance of properly staffing the evaluation team with representatives of the various ICAI development disciplines was pointed out. Procedures for arriving at a slate of candidate products and for gathering comprehensive information about their hardware, software, support, and costs were described. Special emphasis was placed on demonstrations running ICAI-related applications as a means of determining the adequacy of system resources, especially CPU speed and memory capacity. A series of matrices should be constructed to organize technical and cost infromation and to ensure that all products are exhaustively rated on all features of interest. Throughout this process, particular attention must be paid to the information processing limitations and group dynamics of the evaluators so that products are accurately assessed. Several mechanisms for aggregating scores were outlined, and a 100-point scaled rating procedure was proposed as best for most situations.

This evaluation methodology was applied to the selection of an ICAI system for maintenance training research and development. Ten leading AI systems were initially examined, and the six most promising of these were thoroughly studied. Although all six of these candidates obtained the majority of the rating points that could be awarded, all also had substantial room for improvement. None possessed supported software for controlling interactive videodisc presentations, and only one candidate had an authoring system. Broad trade-offs existed between general purpose workstations and LISP machines, with the former tending to excel in hardware and the latter in software. System costs fell substantially during the six-month period of this evaluation, evidencing a general trend that is putting sophisticated AI capabilities within reach of most training organizations.

References

1. Rothkopf, E. Z. (1970) The concept of mathemagenic activities. Review of Educational Research, 40, 325-362.

2. Holzman, T. G. (1988) Computer-based instruction systems: A selection procedure. In A. Kent, J. G. Williams, and A. G. Holzman (Eds.), Encyclopedia of Computer Science and Technology, 17, 30-51. New York: Marcel Dekker.

3. Miller, G. A. (1956) The magical number seven, plus or minus two: Some limits on our capacity for processing information. Psychological Review, 63, 81-97.

4. Stumpf, S. A., Zand, D. E., and Freedman, R. D. (1979) Designing groups for judgmental decisions. Academy of Management Review, 4, 589-600.

MISSION-ORIENTED SIMULATOR DEVELOPMENT

88-4589-CP

Gary R. George - Staff Engineer
Samuel N. Knight - Staff Scientist
Edward A. Stark* - Senior Staff Scientist
Link Flight Simulation Corporation
Binghamton, New York

ABSTRACT

Traditionally, simulators have been specified, developed, and tested for their ability to support objectives which would otherwise be addressed in flight training environments with which crews are generally familiar. Current simulation technology, however, permits additional training support in the form of mission practice, but simulator characteristics relating to realistic combat missions and to the processes of tactical judgement and decisionmaking have not as yet been clearly defined. At the same time, simulator users, concerned about the demands of actual mission conditions on crew proficiency, have begun to employ the simulator in mission training as well as in the training of individual and crew operating skills. Thus, it is not unusual for a simulator to fulfill the requirements of the specification but still fail to meet the training requirements of the user. For this reason it is necessary to supplement the traditional method of simulator development with evaluations oriented toward crew missions rather than toward specific tasks alone.

INTRODUCTION

The operational readiness of military flight crews has been supported for many years through the use of increasingly sophisticated training devices and training systems. Simulation technology has enabled the representation of more and more complex system, mission, and environmental characteristics. Today's training devices are able to replace substantial amounts of practice in real systems, for training in system operation and employment. In addition, however, synthetic training systems are beginning to supplement real systems and real-world environments by permitting essential practice in tasks, missions, and conditions crews cannot experience in real-world training. The simulator substitutes for the aircraft and the training range, and in some cases it replaces them.

In effect, the simulator has become capable of supporting training objectives beyond those derived from the tasks involved in system operation and fundamental tactical operations, permitting practice in the workloads, complexities, and uncertainties typical of real-world tactical operations.

Simulator specifications typically describe the characteristics of the aircraft, its systems, and its operating environment as they are expected to be perceived and interpreted by the crew being trained. It is difficult to specify detailed mission training performance except in general terms, however, because for the most part there is limited analytical data to define the required performance. As qualified crews begin to operate the simulator, the essential characteristics of the mission and task environments are more clearly understood, permitting more detailed and more accurate interpretations of the intent of the specification.

A method of evaluating the simulator's mission training capabilities has been developed and demonstrated in the AH-64 Combat Mission Simulator. It requires that a small team, or teams, of qualified specialists analyze the crew's training requirements and assess system and subsystem designs as they evolve for their potential for supporting the specific training needs of the user. The makeup of the team is crucial, requiring personnel who understand the mission and its training implications, the instructional process, and the implications of specific training and instructional objectives for system design.

The demonstration of this method has shown that it is effective in meeting the specification and in ensuring the support of tasks with which it is concerned, as well as the users' tactical training objectives.

A major side benefit of mission-oriented development is in its effect on simulator cost and schedule. Identifying problems in mission support early in the development process permits designs to be modified immediately, rather than later at excessive cost.

The mission-oriented simulator development process begins with a review of the missions, the tactical capabilities of the systems being simulated and the training objectives derived from those missions and capabilities. Where necessary, specific mission training objectives are expanded to identify their demands for specific simulator capabilities. Mission scenarios are developed and Implemented to exercise these capabilities as they evolve from concept to design, and finally to implementation in the hardware-software integration process. Deficiencies and their likely causes are noted and addressed, and as modifications are implemented they are evaluated in turn. The result is the development of a system which meets specifications and which is optimized to support the user's requirement for combat proficiency.

The mission-oriented development concept as demonstrated in the AH-64 Combat Mission Simulator (CMS) utilized a team of subject matter experts and engineers who employed a series of representative combat mission scenarios to evaluate the performance of the total system and

* Member AIAA

its subsystems throughout the hardware-software integration cycle. The team became proficient in recognizing existing and potential design problems and, through interaction with design engineers and the customer, in finding acceptable solutions to those problems.

Experience in the progressive testing of the AH-64 CMS and its systems suggests that the operational mission must be the primary focus of attention throughout the life cycle of any simulator development program. Requirement analyses, proposal concepts, preliminary and detail designs, system development progress (vs. schedule), demonstration readiness, acceptance test readiness, and similar production unit and follow-on change activities must all be directed at supporting the mission of the operational system. At every point where such validation and testing is applied, the mission team should assess the system from the point of view of the ultimate user, asking such questions as: will this system accomplish our training objectives; will it be adaptable to potential future enhancement; would we buy this system as it is; could it be made better by being designed or developed differently? The mission team seeks an optimized system, leaving final decisions of practicality (cost and schedule) to the development and procurement management teams.

In theory, the evaluation of the simulator for mission training appears to be a relatively simple concept. However, in reality, successful implementation requires a complex integration of expertise and dedication coupled with total acceptance and cooperation by all involved in the program to which it is applied. Team members must be carefully selected to consolidate a total spectrum of expertise, including knowledge of the intended training objectives, technical and operational knowledge of the systems being simulated, and technical knowledge of the simulation designs, including their inherent limitations.

SIMULATOR DEVELOPMENT AND TRAINING OBJECTIVES

Armed helicopter pilots must be proficient in an exceptionally large repertoire of complex skills, and they must be able to exercise them within complex and demanding tactical environments. More important, each skill must be employed within a context of continuous tactical awareness, analysis, judgement, and decisionmaking.

The AH-64 Combat Mission Simulator is one of a number of resources available to the crew in achieving and maintaining the levels of individual, crew, and tactical expertise needed in combat operations. It supports the development and refinement of individual skills and the integration of those skills into proficient crew performance, but its primary function is to provide crew experience in the fundamentals of tactical decisionmaking.

The requirement for the Combat Mission Simulator is addressed in the simulator specification. Specific features are defined to ensure the support of practice in all of the individual and crew tasks performed in combined arms operations. In addition, the simulator specification recognizes that the simulator is to be employed in a Combat Skills training curriculum, to ensure the systematic achievement of all the objectives associated with AH-64 crew tactical performance.

The information needed in learning to fly the aircraft, to operate its systems, and to perform a variety of weapon system and tactical procedures is still relevant and essential, but it is insufficient for the most important functions of the simulator, the development of integrated, interactive tactical skills, and tactical judgement, coupled with the "high stress" environment that today's pilots will experience in actual combat.

Simulator specifications are generally oriented toward the development of features which enable the crew to practice specific tasks. These tasks are taken partly from the flight training curriculum and partly from the superimposition of aircraft and system characteristics on anticipated combat situations. Simulator characteristics are required to support practice in basic normal, emergency, and degraded mode flight and systems operations, and to ensure practice in a range of anticipated adverse flight and operating conditions.

Fundamental task practice can fulfill a variety of training objectives, depending on the quality of the features available in the simulator. Basic task practice, from the point of view of the crew, consists of observing patterns of information requiring specific actions; selecting, coordinating, and executing apparently appropriate responses; evaluating their effect and correcting, as necessary. This level of practice supports perceptual objectives relating to the selection of an appropriate response, and procedural objectives relating to the sequencing and timing of crew performance. Crew coordination and the refinement of psychomotor skills are also supported.

Armed helicopter crews recognize that while practice in discrete tasks and basic tactical procedures is essential in skill development and maintenance, combat readiness requires something more than practice in set exercises. The Combat Skills curriculum reflects this in organizing training within a set of tactical mission scenarios. These incorporate the individual tasks learned in other settings, within a series of mission scenarios representing a number of different tactical problems. The scenarios progress in difficulty, imposing increasingly realistic task loadings, time pressures, and threat effects. The crews learn to interact with each other, with other tactical elements, and perhaps most important, with the terrain and with the threats deployed on and over it. Previously mastered, relatively discrete skills are articulated within increasing levels of workload and stress, and training which began as the integration of discrete combat skills progresses rapidly into the development of fundamental tactical skills.

Tactical skill training requires more of both the student and the simulator than do the individual tasks for which the simulator specification is usually written. Individual tasks, as noted earlier, require that the crew recall a task requirement, select and perform required procedures, and observe their effects on the aircraft, its systems, and tactical environment.

Combat Skills training missions, on the other hand, require the crew to observe information representing complex and ambiguous task settings; to evaluate, prioritize, and interpret that information within the context of the assigned mission; and to select, execute, modify, and rearrange the actions required to maximize the probability of mission success. The value of the Combat Mission Simulator and the Combat Skills curriculum is in enabling the crew to practice interpreting and prioritizing a multiplicity of tactical information, and to generate and evaluate a variety of courses of action in real time. Initial simulator testing is the first opportunity to perform many of the tactical functions assigned to the aircraft. As a result, early system tests serve not only to measure simulator compliance with the specification, but also to explore the aircraft's mission capabilities as well as the simulator's ability to support them. The progressive evaluations that are integral to mission-oriented development simulation explore the tactical capabilities of the aircraft itself, the ability of the simulator to support practice in the tasks to be performed in the Combat Skills curriculum, and most importantly the simulator's ability to support training in unique and complex perceptions and judgements required in tactical operations.

MISSION-ORIENTED DEVELOPMENT CONCEPTS

The basic concept of mission-oriented development is the utilization of a small team (mission team) or teams of highly qualified individuals to assess every aspect of a system (or subsystem) from the viewpoint of the user. Such assessments should be conducted at each stage of the simulator's development and implementation. To be applied as an effective development tool, however, this basic concept must include follow-up activities by the mission team, specifically in resolving problems recognized during each mission assessment.

The simulator's training mission must have first priority at each point in the development cycle, to ensure that the evolving designs are progressing towards a system that will successfully support its training objectives. Early testing allows early identification of mission deficient areas, thus minimizing the cost and schedule impacts involved with implementing required corrective actions.

Much of the process involves physical tests of the system under development, but it also includes a series of dedicated mental exercises. Conducting a representative combat mission with the available system equipment is an example of the first type of testing but the second must be applied before actual systems are available. For example, design reviews are often conducted at subsystem levels. To supplement these individual reviews, mission evaluations must be applied by a team which conceptually integrates the various designs and then assesses the system by visualizing its operation relative to meeting mission objectives.

THE DEVELOPMENT OF AH-64 MISSION CAPABILITIES

The AH-64 Combat Mission Simulator (CMS) is a new, unique, and extremely effective class of Army training device. It extends well beyond flight simulators and weapon system trainers in supporting practice in high-fidelity simulations of both the aircraft and its threat environment. [1,2,3] Its simulation of the AH-64 and its systems allows the crews to become competent in all of the tactical capabilities of the aircraft, while its unforgiving threat environment forces them to learn to fully exploit them. The CMS has been so successful in enhancing the combat readiness of AH-64 crews that in 1985 it was given the Army's Order of Daedalians Award as Weapon System of the Year.

The informal and formal application of mission-oriented development concepts throughout the CMS development cycle was a primary factor contributing to the final mission effectiveness of the AH-64 CMS.

The level of simulation incorporated in the CMS was derived directly from the mission and task descriptions provided as guidance in the procurement specification. Fidelity levels were defined initially in the CMS proposal, but were refined and in some cases expanded in early discussions, after contract award, with customer and user personnel. For example, early in the program a decision was made to incorporate real-time weapon ballistics models. Also, during the design phase the customer helped in developing a faithful mission environment by becoming jointly involved in the design of the interactive threat algorithm.

As subsystems were developed mission-oriented testing was often employed, especially in areas involving sensory perceptions. For example, Army threat experts assisted design engineers in evaluating the classified tones of the APR-39 simulation. Another example was the evaluation of aerodynamics modeling. In this instance mission testing was performed in steps. AH-64 test pilots worked with the system designers to assess and aid in the refinement of the simulator's aerodynamic performance. This effort was followed by a pre-acceptance evaluation by Army pilots.

Mission-oriented testing became formally recognized as a specific development tool during hardware/software integration (HSI). As development progressed, mission teams became extremely proficient in identifying and helping to resolve system problems during HSI. A side benefit of the experience gained by the mission teams was that the teams could be used to perform very realistic missions for demonstration purposes. During such demonstrations the team provided expertise in explaining system capabilities and in answering questions relating to both training system development and the development of tactics and systems.

The importance of mission orientation was most vividly demonstrated when a second CMS to be utilized for in-plant training (IPT) was to be integrated. Mission teams evaluated the system daily. The teams identified problems, led efforts to correct these problems, and provided inputs to prioritize the following day's activities. As a result of this approach, a task which might otherwise have required several months to complete was accomplished In one month. More specifically, the alternate schedule considered at that time to accomplish the same system integration, but utilizing the more conventional approach of successively integrating and fully testing individual systems, involved three to four months of activities.

Mission-oriented testing also influenced the AH-64 CMS acceptance tests. The utilization of both functional and mission tests provided an optimal method to provide detailed checks of key performance factors and to evaluate integrated system performance. It should be noted that the selection of mission tests must carefully consider the fact that overall performance may vary with different evaluation crews flying the aircraft. For example, gun accuracy, which is strongly influenced by the movement of the aircraft, can be readily demonstrated in a controlled functional test where aircraft parameters are frozen but cannot be as easily demonstrated when the aircraft is being flown as part of a mission test. It should be noted that mission-oriented development tends to significantly reduce the overall amount of time involved In preparing for and completing acceptance tests by reducing the number of test runs and test rewrites. Conventionally, tests must often be rewritten and rerun because individual systems appear correct at the functional level but are later found to have discrepancies when operated in a total systems mission environment. The individual systems must then be corrected and the acceptance tests rerun (often requiring a rewrite of the tests). This cycle often repeats, since individual designers may only correct a superficial problem. Mission-oriented development greatly reduces testing, since the majority of integrated systems and mission discrepancies are identified and corrected before the acceptance tests are generated.

It is also interesting to note that when in-plant crew training commenced, simulator design engineers supported the user in learning to recognize mistakes in employing the capabilities of the aircraft, especially relative to the laser-guided Hellfire missile system. Similarly, during customer acceptance engineers worked directly with the customer to quickly identify and resolve discrepancies. In effect, both instances represented the use of a mission team coupling the expertise of the user with that of the designer.

After units were fielded, mission testing proved to be very beneficial in verifying that total system performance was maintained when changes were added to the delivered baselines.

Mission orientation is also relevant beyond the intended training objectives of the CMS. A unique capability resulting from the fidelity of the AH-64 CMS is the ability to evaluate new aircraft systems and new tactics, in the mission context, especially in relation to crew workload factors imposed by the interactive threat.

THE MISSION TEAM

The selection of the mission team members is critical in ensuring successful mission-oriented simulator development. A team made up of personnel having different interests and expertise is essential; the combination of expertise required is unlikely to be found in one individual. Moreover, mission-oriented testing, as a subjective art, profits from multiple ideologies, interpretations, and opinions which tend to increase the cumulative effectiveness of the team. It is essential that the team collectively possess the following expertise and it is desirable, where possible, to have crossovers of such expertise among team members.

First and foremost, there must be comprehensive and accurate knowledge of the system's mission. Such knowledge can only be obtained through discussions with the customer and the user. Obtaining a detailed understanding of the system specification is a prerequisite to conducting such discussions, not a substitute.

Second, there must be knowledge of the systems being simulated and of their operation. Today's military aircraft incorporate sophisticated, state-of-the-art systems which are integrated to provide versatile maneuverability, precise navigation, long-range target recognition, identification and acquisition, and extremely accurate weapons delivery. Somewhat as a consequence of evolving near the limits of technology, these systems often have highly individualized operational characteristics which are very significant in the split-second decisionmaking environment imposed on the aircrews in actual combat.

To facilitate advanced training for such aircraft, high-fidelity simulators, such as the AH-64 CMS, employ simulations, stimulations, and emulations which rival in complexity the actual systems being replicated. Simulation designers must know the aircraft systems as well as, if not better than, the initial designer since the simulated system must not only perform as well as the aircraft systems but must also interface with a simulated environment and be capable of performing simulator-unique functions not generally considered in the aircraft system design (e.g., freeze, rapid initialization, record/replay, etc.). The mission team must have a detailed knowledge of the aircraft system operation through access to documentation, discussions with system developers, test pilots, and aircrews, and, when possible, through hands-on experience.

While it is desirable for the team members to have detailed knowledge of the actual aircraft systems design, this is not always feasible for large systems. In such a case, the team should possess a good conceptual understanding of the system designs and access to individuals with more detailed knowledge. The intent is that the team be able to recognize and identify problems, which, at a minimum, requires knowing enough detail to ask the right questions. Since training devices are frequently developed in parallel with operational systems, problems revealed through mission testing must be immediately checked against the operation of the actual system or the data representing that operation to reverify the accuracy of the data used to design and assess the respective systems.

With knowledge of the intended mission and of the operation of the actual system, the team can assess the operation of the simulated system. However, the role of the mission team also includes helping to isolate and resolve system inadequacies. Thus, the team must have a detailed knowledge of simulation, both of the specific system design being evaluated and of simulation techniques in general.

It should be noted that the mission teams will be evaluating systems which, to the best knowledge of the individual simulation designers, are operationally correct; therefore, the team will be dealing with system operation, the relevance of its fidelity level to mission and training objectives, and its overall quality with respect to the technology being employed.

The team must understand the design approaches taken and the technologies employed so that once a design problem has been identified the team can assist in isolating its source. The team must assess the relevance of the design approach and its method of implementation, its relationships with other systems, and its compatibility with the limits of the technology.

Individuals having multidisciplinary backgrounds are highly desirable for mission testing since they have a stronger appreciation for the limitations, constraints, and interdependencies of various systems and can best assess performance which may be constrained by design tradeoffs between those systems. For example, advanced tactical performance is heavily dependent on the capabilities of the accompanying visual system, which may lead to tradeoffs between tactical performance and visual aesthetics as discussed in an article written by CW4 William Yarlett[4]. Team members should also be familiar with the various technologies and design approaches employed throughout the simulation industry. The team should be open to considering and recommending an alternate approach which may solve a discrepancy at minimum cost or, in some cases, may represent the only available solution.

Another simulation-related requirement is the knowledge of training methodologies, especially in relation to the training elements achievable with various types and levels of simulation. Instructional interactions must also be considered to verify that the instructor will be able to readily control and assess the simulations being evaluated. The benefits of mission-oriented development are not constrained to full-fidelity simulators. Mission teams can also be utilized to assess whether the training value of a lower-fidelity simulation is adequate to support the intended mission or whether an alternate level should be recommended.

It is also helpful if team members are familiar with pertinent factors affecting the procurement, including technical decisions imposed by cost or schedule constraints and plans for future enhancements. These factors should be taken into account in the recommendations generated by the team.

The qualifications of the initial AH-64 CMS HSI mission team provide an example of the expertise required to successfully accomplish mission-oriented simulator development. The first team member was a former helicopter pilot with combat experience and extensive knowledge of helicopter flight, navigation, communication, and weapons systems and of the technologies available to represent those systems for training. He was also an expert in simulator instructional training systems. The other three members were high-level tactical systems engineers with a combined simulation experience of 42 years, including expertise in the design and integration of tactical, visual, and motion systems (including cue synchronization) and possessing extensive overall knowledge in all other major areas of simulation and state-of-the-art simulation technologies.

TEST SELECTION

Test selection is derived by applying the concept of mission-oriented development, employing the system as the user will employ it. The scope of the test, however, will vary depending on the intent. For example, if a comprehensive assessment of the missile system simulation is desired, then a detailed test must be planned to exercise each and every operational mode. However, in another very useful application of mission-oriented testing, that of morning readiness before training, the tests should be representative, assessing each subsystem but not in every conceivable mode. The difference in this case is that the intent is to provide confidence that the simulator is operational. The confidence is obtained during the mission by looking for "changes" from what is normally observed in each day's test. For example, perceiving changes in such areas as the aerodynamic feel, the visual scene, or the weapons accuracy can quickly indicate a problem which in turn could be more accurately identified with a detailed test. Another important consideration is that it may be desirable for the mission test to encompass more than the system being assessed. For example, mission-oriented testing should be performed in preparation for all customer evaluations. In these evaluations it is desirable for the customer to conduct his tests in an environment in which he is accustomed to operating. Therefore, even though the evaluation is planned for only a subsystem (e.g., Doppler system), the mission test should assure that all supporting systems are fully operational (flight training systems, etc.).

UTILIZATION OF MISSION TEAMS

The mission team should be used to provide an expert assessment of a system or subsystem. The team should also assist in isolating and resolving problems detected, by working closely with the design engineers and management. A key to effectiveness is prioritization. The team members, with their knowledge of the state of the current system, the intended operation, and the tasks involved in completing the system, are most qualified to assist management in setting the priorities for resolving problems. Another important aspect of utilization is follow-up, whereby the team remains involved while the problem resolution is determined, implemented, and rechecked. It should be noted that management can also utilize mission teams to check or verify where their program is relative to schedule and completion of the system development.

To fully derive the benefits of mission-oriented development, management and functional engineering groups must accept and become actively involved with the mission teams. Management reluctance to add mission testing to already full schedules must be overcome through the realization that mission-oriented development helps to reduce schedule and cost by optimizing the prioritization of activities and identifying mission-oriented discrepancies early in the development cycle. Ideally mission testing should be a scheduled activity. Skepticism and initial resentment by designers may also be encountered but experience has shown that this is short-lived as these individuals soon realize how much they are being helped by the team. Results bring acceptance.

CONCLUSIONS

Simulators must be developed to meet both the letter and intent of the specification. The specification defines the minimum essential capabilities of the simulator and its systems required to support specific individual and crew tasks. The intent of the specification is to ensure support of all user training objectives. Mission training requires practice in the tasks identified with the specified simulator systems and subsystems, but it also requires practice in processes which are rarely well defined when the specification is produced, especially when the aircraft itself Is still under development. Mission-oriented development can be used throughout the life cycle of the simulator development program to ensure support of the maximum number of training objectives. The process requires a thorough understanding of underlying mission tasks and training concepts, careful selection of mission teams, and dedicated, user-oriented performance by the entire development team.

REFERENCES

1. Yarlett,W.A., Drew,E.W., "Combat Mission SImulation: The Attractive Alternative", Published Proceedings of the Interservice/Industry Trainlng Systems Conference, Salt Lake City, Utah, Nov 1986.

2. Drew,E.W., George,G.R., Knight, S.N., "AH-64 Combat Mission Simulator Tactical System", Published Proceedings of the AIAA Flight Simulation Conference, Monterey, CA, Aug 1987.

3. Drew,E.W., George, G.R., Knight, S.N., "Visionics Simulation in the AH-64 Combat Mission Simulator", Published Proceedings of the National Aerospace and Electronics Conference, Dayton, OH, May 1988.

4. Yarlett,W.A., "Computer Generated Imagery; Aesthetics or Threat Performance", Army Aviation, Jan 1988.

Training/Simulation Environment for Space Shuttle Processing

88-4592-CP

M. J. Wiskerchen*
Stanford University, Department of Electrical Engineering
Stanford, California 94305

C. Mollakarimi**
Lockheed Space Operations Company
Titusville, Florida 32780

Abstract

The Space Systems Integration and Operations Research Applications (SIORA) Program was initiated in late 1986 as a cooperative applications research effort between Stanford University and NASA Kennedy Space Center (KSC) along with its prime operations contractor, Lockheed Space Operations Company (LSOC). One of the major initial SIORA tasks was the application of automation and robotics technology to all aspects of the Shuttle tile processing and inspection system. This effort has adopted a systems engineering approach consisting of an integrated set of rapid prototyping testbeds in which a government-university-industry team of operations personnel, technologists, and engineers tested and evaluated new concepts and technologies within the operational world of Shuttle. These integrated testbeds included speech recognition and synthesis, laser imaging inspection systems, distributed Ada programming environments, distributed relational database architectures, distributed computer network architectures, multi-media workbenches, expert system applications, probabilistic risk assessment modeling, and human factors considerations.

The integrated testbeds provided an environment to design, evaluate, and develop training and simulation tools for Shuttle processing and operations personnel at KSC. These training/simulation tools were defined and determined from the user-driven testbeds involving the integrated prototype system. As each element of the training/simulation system was validated in the testbed, its functional specifications were documented and the system was migrated to the operational mode. This paper will describe the results of the SIORA project efforts as applied to Shuttle tile processing at Kennedy Space Center.

1. Introduction

The Shuttle program, following in the footsteps of Apollo, had its design driven by R & D mission objectives. Only secondary consideration was given to long term operations and maintenance issues and efficient utilization of the Shuttle system. The Shuttle Thermal Protection System (TPS), like other Shuttle elements, was no different and little consideration was given to optimizing the TPS design for operational maintenance efficiency. This has resulted in a TPS whose maintenance program can be characterized as being man-power intensive and time consuming. This is due to the fact that the maintenance program (based on initial development phase specifications and procedures) uses manual techniques for inspection and measurement, mostly paper databases, no networking between pertinent electronic databases, manual scheduling of operational flows and a quality control and reliability program based on a paper information system.

An important part of the SIORA effort was to understand the organizational dynamics involved in evolving the TPS operations from its present labor-intensive state to one in which functionality and operational efficiency and productivity where primary drivers. Although SIORA began as a systems engineering and technology transfer program, it was soon realized that no implementation progress could be achieved without educating the operations "culture". This culture is composed of NASA and its prime operations contractors with work functions ranging from engineering to quality assurance to technology development.

Introducing new technologies and operational concepts into such a culture is constrained by several problems. First, the TPS processing is so labor-intensive due to manual operations that the work force has no time to think about or to attempt to improve the operational system. As budgets and personnel levels continuously decrease, the downward spiral continues and the ability of the system to change and improve decreases. This required a careful assessment of the appropriate systems engineering approach which not only takes into account the engineering and technical questions but also addresses the organizational and "people" questions. The SIORA Program chose an iterative systems engineering methodology, shown in Figure 1., which emphasizes a team approach (design engineers, operations personnel, technologists) for defining, developing and evaluating new concepts and technologies for the operational system. This is accomplished by utilizing rapid prototyping testbeds whereby the concepts and technologies can be iteratively tested and evaluated by the team in a hands-on mode. In addition to the skill mix of the team, it is important to have membership equally represented by the government, industry and university sectors (Figure 2.). This feature of the SIORA teaming is particularly significant in the areas of rapid acquisition and introduction of state-of-the-art technologies. It also assures that the system derived from this process will be commercial viable and maintained in the future.

Although the mutual learning process between the team members was essential and important, the primary driver for the program was to satisfy the functional needs of the operations personnel. In considering the application of automation and robotics to the TPS several important questions must be asked. First, what technology can be applied which will

* Sr. Res. Associate, Dept. of Electrical Engr.

** Project Manager, LSOC

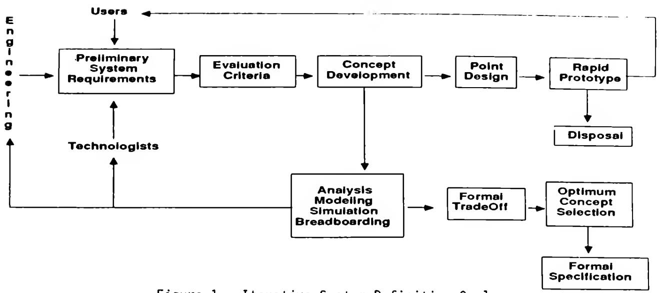


Figure 1. Iterative System Definition Cycle

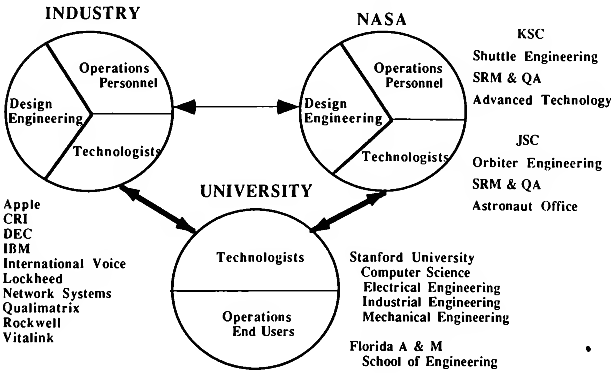


Figure 2. Rapid Prototyping Team

produce significant productivity gains and second, what functional processes and procedures are present which loose their purpose in an automated system? The first question was surprisingly easy to address since all of the technologies were commercially available. We found that the difficult task was the integration of the technologies into an efficient and productive operational system. The first step in identifying applicable technologies was to divide the TPS maintenance system into functional process areas. This produced the following primary areas: multi-media (speech, graphics, imaging systems, text) information capture, distributed computer networks, distributed database architectures, windowed displays, software environment, simulation environment for training, and human factors considerations in system designs. The initial prototype included technologies which addressed each of the above functional areas (Figure 3.). It was also determined that a number of functional processes would be eliminated in an automated system. These revolved primarily around procedures to validate and verify information which resided on paper databases. The interactive electronic system eliminates the need for these activities.

2. System Engineering Methodology

Before starting into the system engineering methodology it is important to establish a general definition for what a system is. The definition we will use is that a system is a complete solution to a defined need in its full environment over its prescribed lifetime. For the SIORA Program, system engineering is then viewed as a process by which user requirements are defined and understood and are subsequently implemented in a system design. The iterative interaction between the users, technologists and design engineers during all phases of a project is critical to make the appropriate transition from perceived user needs to system specifications.

A key element to the system engineering methodology is the formation of the system engineering team. As mentioned previously, a teaming between operations personnel, design engineers, and technologists is essential. Each brings a unique skill and knowledge to the project. The mutual educational interaction of this triad establishes an integrated and

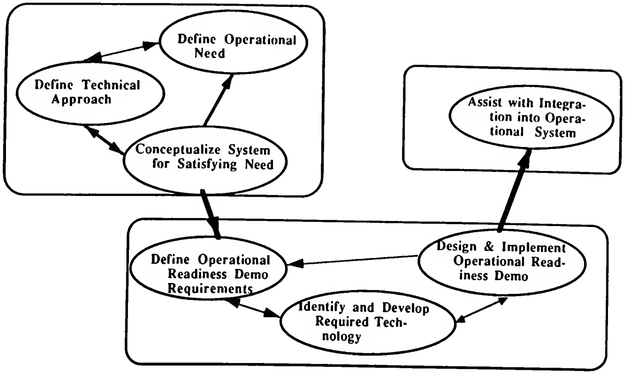
PROTOTYPE TECHNOLOGY		TILE APPLICATION FUNCTION
1. Distributed Relational Databases	↔	ACCESS TO SHUTTLE ENGINEERING AND QA DATA AT KSC, JSC, AND ROCKWELL
2. Local andWide Area Networks	↔	NETWORK INTERFACE FOR ACCESS TO DISTRIBUTED DATABASES, FOR DATA CAPTURE, FOR ELECTRONIC MAIL
3. Multimedia Workstations	↔	MULTIMEDIA INFORMATION INTERFACE FOR SHUTTLE OPERATIONS PERSONNEL
4. Speech Recognition & Computer Speech	↔	TILE DATA CAPTURE AND PROCESS --- PROCEDURE CONTROL
5. Laser Imaging Systems	↔	TILE MEASUREMNT TOOL
6. Ada Software Environment	↔	SOFTWARE ENVIRONMENT DESIGNATED FOR SPACE STATION & SIORA PROJECT
7. Distributed CAD/CAM/CAE Facility	↔	ELECTRONIC SUBMISSION AND ACCESS TO ENGINEERING & QA DRAWINGS
8. Expert System Applications	↔	APPLICATIONS TO SCHEDULING, TREND ANALYSIS, DECISION SUPPORT TOOL
9. Probabilistic Risk Assessment	↔	MODELING TECHNIQUE FOR SAFETY, RELIABILITY, AND QA FOR SHUTTLE

Figure 3. SIORA Technologies & Applications

iterative engineering process resulting in a system implementation which closely tracks the dynamic and evolving user requirements and pertinent technologies. This educational process has several important consequences. First it allows all members of the team to participate in decisions throughout the life cycle of the project. Second, it allows each of the organizational elements (NASA and prime contractors) to stake claim to parts of the project , thus, they feel responsible for its success and its rapid implementation.

Another key element to the system engineering methodology is the utilization of rapid prototyping testbeds in parallel to the ongoing operational systems. These testbeds serve several vital functions. First, they provide an environment which allows the system user/design engineer/technologist triad to obtain quick and unconstrained hands-on experience with new concepts and technology. It is also an environment where design concepts and technology can be modified quickly or discarded if flaws are found. The SIORA Program also emphasizes the importance of having the triad formed out of equal representation from the government, industry and university sectors. Each sector receives unique benefits from its participation in the project. The government sector benefits by being able to evaluate new technologies and concepts outside the formal procurement process without jeopardizing future competitive system procurements. It also has an opportunity to work with university students, the next generation of engineers and scientist, who can be recruited as future government employees. NASA operations personnel, although constrained by schedule, budget pressures and scarce travel funds, have an opportunity to get hands-on exposure to state-of-the-art technologies from industry and the university without having to make long term commitments to that technology. The university sector obtains a rich applications environment to implement and test new ideas and also has a real-world educational environment for its students. Industry benefits in three ways: obtains a high fidelity test environment for its internal R & D; has the opportunity to recruit personnel from the student participants; and establishes a means to better understand the system needs/requirements for future government directed systems.

The final aspect of the methodology is how the results of the prototyping is integrated into the operational environment. For automating and/or upgrading existing systems, the process is carried out in the following way (Figure 4). In the initial stage the prototyping team (operations personnel, design engineers, technologists) identifies the operational functions of the system. The technologists will then identify applicable technology for each of the system functions. This will then be iterated with the system users and prototype design engineers to determine the design options for the system. Some options can be tested with high fidelity computer simulations while others may have to be fabricated into bread- or brass-board prototypes for evaluation by the team. Test and performance criteria are established and agreed to by the prototyping team. In addition, system user productivity gains and cost reductions are carefully evaluated and documented. The primary objective is to rapidly iterate on the prototype until user productivity and cost reductions are at an acceptable level. Since a small cadre of operations personnel have been participating in the prototyping process, the transition of the prototype final design has, in essence, been initiated. The functional specifications derived from the prototyping process is then formally documented and used as inputs to a competitive procurement process for the new operational system. At this same point in time, considerable effort must be spent by the prototyping team to develop off-line prototype training modules to educate and train operations personnel. The new and old systems must be operated in parallel until new prototype system elements have been integrated and validated and the operations personnel fully trained.



Define Operational Need
Define Technical Approach
Conceptualize System for Satisfying Need
Assist with Integration into Operational System
Define Operational Readiness Demo Requirements
Design & Implement Operational Readiness Demo
Identify and Develop Required Technology

Figure 4. SIORA Systems Engineering Approach

3. Tile Automation Project - Theory Meets Reality

The above discussions on iterative systems engineering and rapid prototyping in operational environments is only a theoretical model until it actually gets implemented in an operational environment. At the beginning of the project, it was jokingly stated that we should rapid prototype the rapid prototype project before we actually addressed a critical operational system on Shuttle. Since that wasn't possible, we proceeded with the tile automation project for real and kept a flexible attitude toward the system engineering methodology and approach. It is important to recount the "lessons learned" to better understand and quantify the methodology so that it can be successfully applied in other operational environments.

One of the first areas to discuss is the various organizational perceptions at the start of the project. From a NASA point of view, this brought together three diverse organizational entities, Shuttle Engineering and Operations, Safety, Reliability, Maintainability & QA, and Advanced Technology. Each is driven by different goals and responsibilities and by different schedules and budgets. Historically, Shuttle Engineering & Operations and SRM & QA have had a working engineer - inspector relationship rather than a team attitude.· The advanced technology group has traditionally been considered "sand box types" where technologies are developed that, although designed to address a perceived functional operational requirement, have low probability of ever being implemented into the operational environment.

From the operations prime contractor (Lockheed Space Operations Company (LSOC)) point of view, the project was initially looked at as an interesting concept with some potential to give long term relief from the labor-intensive tile processing problems. Historically, as part of the prime operations contract, contractors do not have contractual authority to pursue studies for implementing new operations concepts or technology. In addition, the present budgetary, man-power level and schedule climate (post-Challenger) in the Shuttle processing environment made LSOC very conservative in its expectations for the project. The initial response from LSOC was to dedicate several personnel to the project which had considerable experience with Shuttle tile processing but were not part of the flow schedule for Orbiter 103, Discovery.

Stanford University was also playing a historically non-traditional role. Stanford saw the SIORA Project as a means to provide their students and faculty with an applications environment to test and evaluate systems engineering techniques and newly developed technologies. Although close university-industry-government ties for cooperative research is not new to Stanford, applications research in an operations environment is. Providing a "real" systems engineering educational experience for students within the Shuttle program is also new to the School of Engineering at Stanford. The cooperative agreement between KSC and Stanford was also the first of its kind at KSC. This agreement allows KSC and Stanford to jointly share personnel and facilities and also closely coordinate and manage the joint project. The agreement also allows industrial partners of Stanford to participate on a cost sharing basis on the rapid prototyping efforts. This feature of the program allows the placement of state-of-the-art prototype equipment (loaned, gifted or heavily discounted to Stanford) in the middle of NASA operations without violating or jeopardizing future competitive procurements to acquire the operational system.

With the above participants in place, the tile automation project was ready to begin. It is important to realize that all Shuttle tile processing operations or potential operations comes under the management and review of the Shuttle TPS review boards. There are three boards, Level 3, Level 2, and Level 1. The Level 3 and 2 boards are composed of JSC and KSC personnel with Rockwell and Lockheed Space

Operations Company (LSOC) as support personnel. Level 1 is a policy level board and is operated out of NASA HQ. To initiate the tile automation project, Stanford and LSOC submitted a project plan to the Level 3 board which subsequently approved it and passed it to the Level 2 board. With Level 2 approval, funds were allocated from the Shuttle operations budget to prototype, test, evaluate and finally specify the functional configuration for the operational system. The project plan layed out a scheduled for a 15 month period for completion of the prototype evaluation and for functional specifications to be documented. A follow-on period of nine months was allocated for the competitive procurement and, in parallel, to develop training models and simulation capabilities for the operational system.

A severe constraint appeared immediately. This was in the area of network communications between the geographically distributed, both locally and outside of KSC, functions in the program. Although the computer equipment was delivered within a month of Level 2 approval, basic infrastructure resources like cabling for the computer networking and computer room utilities (power, air conditioning) were not available for 6 months. These delays were primarily due to the generally perceived low priority given for the project by mid-level operations management. Despite these handicaps, considerable progress was made in defining and developing the individual prototype elements of the project.

One of the more important architectural decisions made in the initial phase of the program was the selection of an Ada software environment. Although NASA had already made the decision on Ada for the space station core software environment, no one had attempted to put Ada into an operational Shuttle program. With a crash training program in Ada at Stanford, the prototyping team developed the programming skills quickly. Our efforts were aided by the recruitment of a software company, CRI, Inc., which provided a relational database system which was programmed in Ada and easily ported to a number of different computers including DEC and IBM. Our experience on the project has shown that Ada provides a good software environment for quality and quantity of code while remaining hardware independent. Since we were only prototyping, we decided not to investigate configuration managed Ada environments. This turned out to be a mistake when the prototype was pressed into operation for work on Discovery. The prototype team is now being added to by industry partners with expertise in Ada software environment tools including tools for distributed programming and configuration management.

Another important architectural decision was the implementation of a distributed network configuration. This is a concept where nodes on a network each serve a specific function. This is opposite to the "mainframe" concept where a large mainframe carries out a number of functions in a centralized location. The distributed concept takes advantage of the decreasing cost of specialized hardware for unique functions. This allows for high performance, good access from anywhere on the network, and a graceful, both in terms of budget and functionality, evolution as technology evolves. The distributed network concept also allows for the tailoring of workstations to meet the unique needs of classes of operations personnel. The workstations can be developed in a modular manner which can respond to the particular functions of each job. This makes the task of creating training/simulation system much easier.

The two architectural decisions above allowed the team to progress rapidly. Our success became very visible which, in turn, lead to NASA management requesting that we modify our objectives and schedules. The opportunity was presented to the team to join the operational personnel to assist in processing the orbiter, Discovery. This is quite a challenge for a research/prototyping team. Questions arose as to whether we could get the prototypes certified for use, whether we could meet such condensed schedules, and whether this would drastically interfere with our initially assigned responsibilities. We decided to accept the challenge. We first assessed which of the prototype elements would best aid the work flow for Discovery. It was determined that the speech recognition system and the relational database were the only ones that could make an impact. Through hard work and dedication from the entire team the systems were delivered to the tile processing personnel working on Discovery. To our surprise, we found minimal acceptance of our labor saving tools. This was later determined to be due to the fact that none of the work force had an opportunity to get hands-on experience with the tools and, therefore, had no confidence in its use under time critical constraints. With that lessen learned, we are now acquainting the Atlantis (Orbiter 104) tile processing personnel with the prototype system in a hands-on manner. Although the excursion into the operational world delayed our original schedule, the information gained from the effort has convinced the team that this should be part of the systems engineering methodology for upgrading operational systems. To do it successfully, rigorous detailed scheduling is essential and appropriate allocation of resources (time and personnel) should be provided.

4. Conclusions

The SIORA program has been a novel experiment. It has been unique in its approach and its results. Throughout its short existence, the program has attracted a number of creative people from industry, NASA, and the university. These people have focussed on a common goal of improving the operational environment of the Shuttle. Significant progress has been made in this area and by 1989, the new system will be in operation. More important, is the realization that a new systems engineering methodology has been tested and validated. The team will continue to improve on the methodology and quantify its structure so that it can be applied elsewhere.

88-4593-CP

A COMPUTER SYSTEMS UPGRADE FOR THE SHUTTLE MISSION TRAINING FACILITY

Ankur R. Hajare and Patrick M. Brown
MITRE
Houston, Texas

Abstract

The Shuttle Mission Training Facility at the NASA Johnson Space Center is used to train astronauts for Space Shuttle missions. It also supports training of ground controllers in the Mission Control Center. This facility is undergoing a thorough upgrade that will take a few years to complete. The upgrade is divided into four steps, the first of which is currently in progress. Step I consists of replacing the old computer systems with compatible current generation computer systems. The more powerful Sperry 1100/92s are replacing the old Univac 1100/44 host computers. Each of the three new Concurrent Computer Corporation 3280s is replacing three Perkin-Elmer 8/32s plus a Perkin-Elmer 3250. Along with the upgrade in computer hardware, there is a change in operating systems and in the software architecture of the simulators. This paper describes the design alternatives and the problems encountered with the upgrade. The problems include software conversion, support for ongoing operations, and interfacing the new computers to the old shared memory system. The phased implementation methodology of the upgrade is presented along with plans for the future.

Introduction

The Shuttle Mission Training Facility (SMTF) at the NASA Johnson Space Center is the primary facility for training astronauts for Space Shuttle missions. Training support for the Space Shuttle Program started with the building of the Orbital Aeroflight Simulator in 1977 to support the approach and landing tests conducted with the Enterprise. The Shuttle Procedures Simulator (SPS) was built to support the development of procedures performed by the astronauts and to provide generic (not mission specific) training. The Shuttle Mission Simulator (SMS) was subsequently built to provide mission specific training. Astronaut training in the SMS started in 1979, well in time to support the first Space Shuttle flight in 1981. The SMS consists of a Fixed Base (FB) simulator and a Motion Base (MB) simulator. Ascent and entry training is performed on the MB whereas on-orbit training is performed primarily on the FB.

The Network Simulation System (NSS) was added to support training of ground controllers in the Mission Control Center (MCC). The NSS provides communications between a SMS base and the MCC, simulating communications between the Space Shuttle and the MCC. The SpaceLab Simulator (SLS) was built to train astronauts for activities performed in the European Space Agency's SpaceLab which is carried in the cargo bay of the Space Shuttle. The SLS can operate either stand-alone or in combination with one of the SMS bases. In 1981 the SPS was replaced by the Guidance and Navigation Simulator (GNS). The GNS is a Space Shuttle simulator with less capabilities than the SMS. The GNS serves as a test-bed for new simulator software and hardware. Besides these simulators, the SMTF contains the equipment and systems to develop new capabilities and to maintain and operate the simulators.

Due to its age, the SMTF has been experiencing problems [1] which include obsolescence, shortage of capacity, low reliability and poor maintainability. Because of these problems, NASA and its contractors performed studies for upgrading the SMTF and proposed an ambitious upgrade configuration [2] and a design that would overcome the above mentioned problems of the SMTF. Because of budget constraints and facility constraints, it was not possible to implement such an ambitious concept immediately. Instead, an upgrade plan consisting of four steps was formulated.

Step I, which is described here, consists of replacing many of the computers in the SMTF with bigger current generation machines. This will increase computational capacity and will improve reliability and maintainability.

Former Configuration

The configuration of the SMTF before the upgrade is shown in Figure 1 and is described briefly here. A more detailed description can be found in [3] and [4].

The three Space Shuttle simulators within the SMTF (FB, MB, and GNS) are not identical since they are used for somewhat different purposes. However, the primary computing center of each simulator contains identical computer systems and executes the same software loads. Each simulator has a host computer and four intelligent controllers (ICs) as shown in Figure 1.

The host computers for the MB, FB and the GNS are Univac* 1100/44s. The host computer performs the main computational work required for the real-time flight simulation including computation of the equations of motion and executing simulation models for the various subsystems of the Space Shuttle. The 1100/44 host computer has four processors and two Input/Output Access Units (IOAUs).

The Intelligent Controllers (ICs) provide a smart interface between the host and the simulation hardware. The original ICs were Interdata** 8/32 super-minicomputers interconnected by means of shared memory. An I/O IC was added in 1982 to provide more capacity for transferring data between the host and IC shared memory. It is a Perkin-Elmer 3250, which was the highest performance member of the family at the time of its acquisition. The ICs are connected to the host by channel-to-channel interfaces. This interface, known as the "Chi interface", was custom built for the SMS. The ICs are 32-bit machines whereas the host is a 36-bit computer. The Chi interface performs bytes-to-word and word-to-bytes format conversion with a choice of 16 formats under program control.

The ICs include the Crew Instructor Operator Station (CIOS) IC which interfaces the Instructor-Operator Station (IOS) and the simulator crew station to the host. The interface between the CIOS IC and the simulator crew station is via Signal Conversion Equipment (SCE) that performs analog-to-digital and digital-to-analog conversions. The Visual System (VIS) IC is the interface between the host and the visual system that generates the out-the-window scenes for the crew station. The VIS IC also performs several other functions. Consequently, the GNS has a VIS IC even though it does not have a visual system.

The real-time flight simulation runs the Space Shuttle's on-board computer system, consisting of five IBM General Purpose Computers (GPCs), and executing actual flight software. A custom built device, called the Simulation Interface Device (SID) houses the GPCs and interfaces them to the rest of the simulator. The SID interfaces to the host through both the SID IC and the VIS IC. The SID IC controls the operation of the SID and performs the data format conversion between the IBM 32-bit format used by the GPCs and the Univac 36-bit format.

The Payload Simulator (PLS) was added because the host did not have sufficient capacity for high fidelity simulation of complex Shuttle payloads. It is a PE 3200 MPS that is interfaced to the shared memory of the ICs. The PLS cannot run stand-alone; it can only run in combination with the Space Shuttle simulator.

The NSS contains three PE 8/32s and the SLS has two PE 8/32s. The visual system of the MB simulator contains one PE 8/32 to support the forward displays and the visual system of the FB simulator contains four additional PE 8/32s to support the aft and overhead displays.

The interconnection between the 8/32s consists of multiple shared memories. The three 8/32s of the NSS have their own shared memory. In addition, the NSS can access the shared memory of the ICs of either the FB or the MB simulator. Similarly, the two 8/32s of the SLS have their own shared memory and each one can access the shared memory of the ICs of the FB or MB simulator. The visual systems of the FB and the MB simulators are connected to the respective VIS ICs via shared memory.

A switch enables either of the two 1100/44 hosts in the SMS to be connected to either the FB or the MB. The ICs are connected directly (i.e. without a switch) to the crew station and other simulation equipment. Thus, the switch is between the host and the four ICs. The switch allows simulator operations to be resumed on a desired base if one host goes down.

The SMTF contains computer systems to support development and operation of the simulators. These include a PE 3250 development system, a Sperry 1100/91 development system, an IBM 4381 reconfiguration system, and a visual database generation system.

The operating system used on the 1100/44 hosts is the Series 1100 Executive System, commonly known as Exec. This is the vendor's commercial off-the-shelf (COTS) operating system. Level 37, the last version capable of running on the 1100/44, is used. The real-time flight simulation progresses in 40 ms "frames", i.e. at 25 frames/second. Simulation software on the host is organized into "frame jobs" which are executed at iteration rates of 25 Hz or submultiples thereof. Since the standard Exec does not support this type of task dispatching, it was modified to do so. A locally developed extension, called the Frame Time Dispatcher (FTD), schedules and dispatches the frame jobs for execution on the four processors of the host.

The operating system used on the ICs during real-time flight simulation is a Real-Time Monitor (RTM) developed and maintained by Singer [5]. At the time the project started, the vendor-supplied operating system was deficient in several areas critical to real-time operation.

* Later called Sperry, and now known as Unisys.

** Later called Perkin-Elmer, and now known as Concurrent Computer Corporation.

Therefore, RTM was developed specifically for flight simulation and contains only those features required to support it.

Simulation software on the ICs is organized into jump lists or sequences which are executed at iteration rates of 25 Hz or submultiples thereof. An iteration rate of 12.5 Hz is achieved by executing a jump list every other frame; a rate of 6.25 Hz is achieved by executing a jump list every fourth frame, and so on.

Computer Systems Alternatives

Step I of the SMTF upgrade consists of replacing the host computers and the ICs of the three Space Shuttle simulators. The alternatives for the replacement computers were (1) compatible computers, and (2) non-compatible computers. As required by federal procurement regulations, a conversion study was performed in order to estimate the cost of conversion to both alternatives. The compatible alternative for the host computer was the Sperry 1100/90 system, which is the latest addition to the 1100 family and its highest performance member. It is fully upward compatible with the 1100/44 at the application program level. The compatible alternative for the ICs was the Concurrent Computer Corporation 3280, which is the latest high-performance member of the 3200 family. The non-compatible alternative chosen for the purpose of conversion cost comparison consisted of an IBM 3081 host and a Gould 9780 IC.

The conversion cost model prescribed by the Federal Conversion Support Center [6] was used for the conversion study. Using this model, the conversion effort for host and IC software was estimated at 63 staff years for a compatible upgrade and 245 staff years for a non-compatible upgrade. NASA consequently chose to acquire compatible computers for the SMS and GNS upgrades.

Computer Systems Sizing

A study performed for the acquisition of the Sperry development system [7] indicated that a 1100/91 (i.e. a 1100/90 system with one processor) is more powerful than a 1100/44. Thus, a 1100/91 may have been capable of replacing the 1100/44 host. However, the study addressed performance for a development system, not performance for a real-time flight simulation. Previous benchmarking with the SMS workload on Sperry systems [8] had demonstrated performance ratios significantly different from those derived from generic workloads. Thus, there was a risk that the 1100/91 would fall short of the necessary performance. Furthermore, a 1100/91 would provide little room for growth and would not alleviate the capacity problems that existed.

A sizing analysis [9] estimated that a 1100/91 would be 64% to 76% busy with the existing simulation workload whereas a 1100/92 (i.e. a 1100/90 system with 2 processors) would be 34% to 44 % busy with that workload. Thus, a 1100/92 would provide capacity for the anticipated growth of simulation software. Therefore, a 1100/92 was the choice for the new host.

The new 3280 is a multiprocessor system that can have up to 6 processors, including one Central Processing Unit (CPU) and a combination of Auxiliary Processing Units (APUs) and Input/Output Processors (IOPs). The CPU was estimated [10] to be four to five times as powerful as one 8/32 or roughly three to four times as powerful as a 3250. Each APU in the 3280 is as powerful as the CPU. Thus, one multiprocessor 3280 was clearly capable of replacing all four ICs with room for growth and room for error in performance assessment.

The use of one IC to replace four has multiple advantages including an improvement in reliability, which is a major goal of the upgrade. The use of one IC simplifies software because data transfers between ICs are eliminated and also simplifies simulator operations. Finally, the hardware cost of one multiprocessor IC is less than the cost of four uniprocessor ICs. A sizing study [9] recommended a configuration containing four processors and estimated that it would be utilized 47% to 54%.

Upgraded Configuration

The planned configuration of the SMTF after Step I of the upgrade is shown in Figure 2. A comparison of Figure 1 and Figure 2 shows that a 1100/92 replaces each 1100/44 host. The 1100/92 host in the GNS also serves as the software development system for the SMS instead of the 1100/91 development system in the former configuration. A 3280, called the base IC, replaces the following four ICs in the old configuration: CIOS IC, VIS IC, SID IC and I/O IC.

Interfacing/Connectivity

A goal of the SMTF upgrade is to replace custom-built products with COTS products where practical. Therefore, an alternative to the custom-built Chi interface was investigated. The only COTS product that was available was a block multiplexor channel interface made by PE. This was an old product developed to interface 8/32s to IBM 370s. It can be used with a 1100/90 system which can be equipped with an IBM compatible channel. This interface had a maximum rate of 500 kilobytes per second and did not perform any format conversions. Therefore, the format conversions performed by the Chi board would have to be done in software and that would levy a significant load on the host.

Concurrent Computer Corporation (CCC) was developing a new interface, with a speed of 3 megabytes per second, to connect the 3280 with IBM mainframes. One such interface could replace all the Chi interfaces on one simulator. However, CCC did not have a firm date for the delivery of this new product. Therefore, it was decided to retain the Chi interface.

Figure 2 shows that the new configuration still has several 8/32s. The new base IC interfaces to the 8/32s in the NSS, SLS and the visual system. In addition, it interfaces to the PLS, which is a 3200 MPS. The old ICs interface by means of shared memory. Thus, it was necessary to connect the new 3280 to the shared memory of the old 8/32s. Since a COTS product was not available for this purpose, a custom interface had to be designed and built.

Operating Systems & Software Structure

Two operating systems are used on the PE computers in the SMTF. They are: (1) OS/32, the vendor's COTS operating system; and (2) a locally developed and maintained real-time monitor (RTM). RTM is used on all the 8/32s and 3250s during real-time simulation. OS/32 is used on these computers for activities other than real-time simulation. The PE 3250 development system runs OS/32 exclusively.

When the multiprocessor PE 3200 MPS was selected for the PLS, both RTM and OS/32 were considered for real-time use. Since RTM runs only on uniprocessor machines, the alternatives were to (1) enhance RTM to run on a multiprocessor system, or (2) adapt and augment OS/32 for use in real-time flight simulation. A study [11] recommended the use of OS/32. Accordingly, OS/32 is used on the PLS and real-time support software has been developed for it. Since the 3280 is upward compatible with the 3200 MPS, the support software for running OS/32 on the base IC was already available.

In view of the above, OS/32 was the clear choice for the operating system on the new base IC. However, this required conversion of existing IC software to run under OS/32 instead of RTM. The structures of I/O device drivers in OS/32 and RTM are totally different. Therefore, new device drivers had to be written for all the simulation devices connected to the ICs. In addition, the software jump lists in the four old ICs had to be combined and restructured into new jump lists. Functions that were replicated in each of the four ICs were consolidated into the new base IC.

The operating system on the old hosts is Exec Level 37 which was the last version to run on the 1100/44. The version that runs on the 1100/90 systems is Level 39. Therefore, upgrading the host required a transition to the new level of the operating system.

The old frame job structure was tailored to the host configuration, specifically to four processors operating concurrently on the real-time simulation. The new host has two processors. Therefore, the software structure had to be changed. To further the goal of using COTS products, FTD has been replaced by a new dispatching scheme which utilizes enhanced capabilities existing in the new operating system.

Implementation Methodology

The most stringent constraint during implementation of the upgrade was the need to support crew training. Floor space and budget limitations did not permit full parallel operation of all the old and the new computer systems. Therefore, a phased implementation scheme was developed.

When Step I started, the 1100/91 development system was used to code and test the new host dispatching scheme as well as the software changes to the new version of the Exec. For the IC implementation, the 3250 development system was used to verify portability and to develop the new support software.

The 1100/91 was expanded to a 1100/92 by the addition of one processor. Also, additional memory, I/O channels, a string of disk drives and an I/O processor were installed, thereby making the computer system capable of supporting both real-time simulation and software development. Then, a 3280 was installed in the GNS. A system of switches was installed to permit operation with the old and new computers. Thus, old simulation software loads to be used absolutely unchanged while the software is being revised and tested. This interim configuration of the GNS will be retained until the upgrade is completed in June 1988.

The GNS is not identical to the SMS, with the main differences being the visual systems in the SMS and the connection to the NSS and the SLS. Therefore, even after the new system is tested in the GNS, a considerable amount of test and verification must be performed in the SMS concurrently with crew training for Shuttle flights. This constraint necessitated two interim configurations for the SMS.

The two interim configurations are illustrated in Figures 3 and 4. The first interim configuration (Figure 3) retains all the old computers but has one 1100/92 and 3280 pair which is switchable to either the FB or the MB. Thus, training can proceed on any one base while the new computers are being checked out with the other base. The NSS and the SLS can be connected to either the old or the new computers.

To arrive at the second interim configuration, one set of the old computers is removed and a second set of new computers is installed. As shown in Figure 4, the second interim configuration has one 1100/44 host and one set of old ICs. These can be connected to either the FB or the MB. This permits the use of old software on either base and this configuration will be retained as long as there is a need to train with old simulation software loads.

Software Conversion and Testing

Since the Step I upgrade is a compatible computer replacement, it would appear that application software merely has to be recompiled and relinked to run on the new hardware. However, this seemingly simple software conversion is complicated by two factors in addition to the change in operating systems and frame job structure discussed earlier. First, the software is continually being modified and, second, the conversion to the new computer systems must be tested and validated to the satisfaction of various parties.

The Systems Development Division (SDD) is responsible for the upgrade which is being performed by SDD's development contractor. The Facility and Support Systems Division (FSSD) is responsible for the operation, maintenance and sustaining engineering of the SMTF. FSSD's operations contractor continually modifies the "baseline" simulation software from which individual simulation loads are built for each mission. Changes to the baseline are required because the SMTF software must track engineering changes to the Space Shuttle. Besides, each new Shuttle payload requires new software. In this context, it is not permissible to freeze all software through the conversion process, as is usually recommended [6]. Therefore, the development contractor was given the baseline and four simulation loads which included satellite deployment and SpaceLab simulation. The development contractor is responsible for converting these four loads and the baseline and for formally testing and validating them. Furthermore, the development contractor is responsible for incorporating all software modifications to the baseline that occur during the project.

The operations contractor is responsible for all new loads generated from the converted baseline. However, such new loads cannot be generated in time to provide training for the upcoming Shuttle flights. Therefore, the crews assigned to STS-26 (the next flight), STS-27 and STS-28 will receive their mission specific training on the old systems. The crew assigned to STS-29 will be the first one to train with loads generated from the converted baseline. The four loads converted by the development contractor will be used to provide proficiency (i.e. generic, not mission specific) training to other astronauts.

The converted software will undergo a series of tests. First, the development contractor must test it to the satisfaction of the SDD. Next, the conversion will undergo acceptance testing by FSSD and its operations contractor before they assume responsibility for operating the upgraded systems. Finally, the converted software must be demonstrated to the satisfaction of the Training Division who are the users of the SMTF.

Initially, the off-line support software must be shown to retain the same capability. This will be accomplished in a series of stress tests and formal acceptance tests. Next, the real-time simulation software must be demonstrated during launch, ascent, on-orbit, and descent phases of a mission. Critical flight parameters will be logged during a simulation run on both the old and new computer system and then compared for agreement. The final tests, scheduled over a month, will have astronauts actually fly the simulators to establish their confidence in the performance of the new system.

In these tests the upgraded system is required to perform as well as the old system, but not necessarily better. Since this is a hardware upgrade, direct improvements in fidelity of the simulation models are not expected, and any preexisting problems discovered during testing lie outside the scope of this project. With the greatly improved performance and capacity of the new computers, there may be indirect improvements in fidelity that simply result from better response time.

Future Plans

Step II of the SMTF upgrade will continue the goal of replacing obsolete computer systems and providing more capacity to accommodate simulation fidelity enhancements. All the remaining 8/32s will be eliminated in Step II. However, Step II is more than just a compatible computer replacement. The new systems will not only improve fidelity and reliability, but will facilitate a simplification of the system architecture and the operational procedures.

There are four projects that comprise Step II, three of which will replace all the remaining 8/32s. The fourth project is a replacement of the simulated Shuttle voice communications system with upgraded equipment that includes components from the actual operational system.

The highest priority project in Step II is replacing the visual system on the MB and the FB. When NASA purchased this digital image generation system in the late seventies, it was the first production model on the market and represented the state of the art. Although the system has been enhanced beyond its original capability, current image generation and display technology is far more advanced. The current database of visual scenes, which includes all the Space Shuttle landing sites, will have to be converted to the new system, but with modern tools this is expected to be a relatively easy off-line task.

The visual system upgrade will result in the replacement of the six 8/32s in the current visual systems. Since the new visual systems will be acquired by an open competition, the type of computers that replace the 8/32s is not known. Depending upon the brand of the new visual system, the Visual Database Generation System (Figure 1) in the development environment may have to be replaced.

The second computer replacement of the Step II Upgrade will be the three 8/32s in the NSS. These computers will be consolidated into one computer that maintains commonality with the current architecture. A software conversion study will be performed to compare a compatible upgrade versus a non-compatible one. In addition to the computer upgrade, all the specialized communication equipment used to simulate the uplink and downlink telemetry streams, digitized voice, and air-to-ground links will be replaced with modern equipment that provides improvements in capacity and fidelity. One important subsystem upgrade of the NSS project will be a replacement of the console displays in the instructor station with modern workstations. This subsystem will later be used to upgrade the instructor station on the FB and the MB.

A secondary benefit of the NSS upgrade project will come indirectly through its implementation. To minimize the impact on training, the new system will be tested by making it the link between the GNS and the MCC. (Currently, the NSS links the FB and MB to the MCC, but does not link the GNS to the MCC.) This link is a new capability that will allow future, integrated simulations (i.e. mission simulations involving the participation of the MCC) to be tested in the GNS and thus make more time available for training on the MB and FB.

The third computer upgrade in Step II is a replacement of the two 8/32s in the SLS. The SpaceLab has been recognized as another, although specialized, payload which should be simulated in the same computer environment as the other payloads. A PE 3200 MPS, originally acquired for the Centaur project, has been proposed as a replacement for the two 8/32s. Use of this computer system, along with the real-time operating system and support software used for the payload simulation, will achieve the goal of a common environment.

The requirements for Step II are presently being reviewed by NASA and the projects discussed above may be revised for budgetary reasons.

References

1. Hajare, A. R. and Brown, P. M. "Upgrading the Shuttle Mission Simulators", Simulators V, Simulation Series, Vol. 19, No. 4, The Society for Computer Simulation, San Diego, California. 1988.

2. Cockrell, A. A., Gorski, A. M., and Wood, W. B. "Operational Concept of the Upgraded Shuttle Mission Training Facility (SMTF), JSC 22089, MITRE, Houston, Texas. 1986.

3. Hajare, A. R. "Overview of SMS Intelligent Controllers", JSC 17277, MITRE, Houston, Texas. 1981.

4. Brown, P. M. et al. "Upgrade Study for a New Architecture for the Shuttle Mission Simulator", JSC 19474, MITRE, Houston, Texas. 1984.

5. Hajare, A. R. and Dias, L. A. "Perkin-Elmer 8/32 Operating Systems Review", JSC 18303, MITRE, Houston, Texas. 1982.

6. Federal Conversion Support Center Conversion Cost Model (Version 2), Report No. OSD/FCSC-82/001, Office of Software Development, Falls Church, Virginia. 1982.

7. Anderson, S. V. and Brown, P. M. "Review of the GNS Workload and the Sperry 1100/70 Proposal", JSC 19647, MITRE, Houston, Texas. 1984.

8. Hajare, A. R. "Benchmarks for a Computer System for NASA's Shuttle Procedures Simulator", AIAA Flight Simulation Technologies Conference, American Institute of Aeronautics and Astronautics, New York. 1983.

9. Wood, W. B. and Hajare, A. R. "SMTF Computer Upgrade: A Compatible Option". Briefing to NASA (private communication), MITRE, Houston, Texas. 1986.

10. Hajare, A. R. "PE 8/32 Replacement Alternatives". Briefing to NASA (private communication), MITRE, Houston, Texas. 1986.

11. Mitchell, R. C. "Payload Simulator Upgrade: Considerations on the Options", JSC 19695, MITRE, Houston, Texas. 1984.

Acknowledgement

This work was sponsored by Air Force contract number F19628-86-C-0001 T8114M.

List of Abbreviations

APU	Auxiliary Processing Unit
CCC	Concurrent Computer Corporation
CIOS	Crew/Instructor/Operator Station
COTS	Commercial Off-the-Shelf
CPU	Central Processing Unit
FB	Fixed Base
FCSC	Federal Conversion Support Center
FSSD	Facility and Support Systems Division
FTD	Frame Time Dispatcher
GNS	Guidance and Navigation Simulator
GPC	General Purpose Computer
Hz	Hertz (cycles per second)
IBM	International Business Machines
IC	Intelligent Controller
I/O	Input/Output
IOAU	Input/Output Access Unit
IOP	Input/Output Processor
IOS	Instructor/Operator Station
JSC	(Lyndon B.) Johnson Space Center
MB	Motion Base
MCC	Mission Control Center
ms	milliseconds
NASA	National Aeronautics and Space Administration
NSS	Network Simulation System
PE	Perkin-Elmer
PLS	Payloads Simulator
RTM	Real-Time Monitor
SCE	Signal Conversion Equipment
SDD	Systems Development Division
SID	Simulation Interface Device
SLS	SpaceLab Simulator
SMS	Shuttle Mission Simulator
SMTF	Shuttle Mission Training Facility
SPS	Shuttle Procedures Simulator
VIS	Visual System

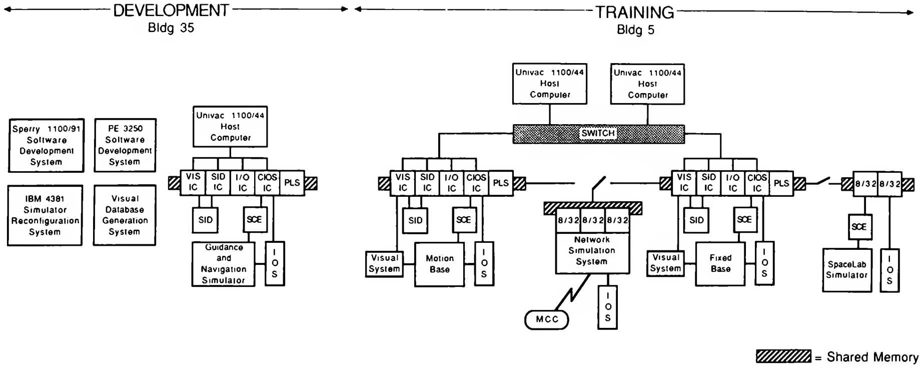

DEVELOPMENT
Bldg 35
TRAINING
Bldg 5
Sperry 1100/91 Software Development System
PE 3250 Software Development System
IBM 4381 Simulator Reconfiguration System
Visual Database Generation System
Univac 1100/44 Host Computer
VIS IC
SID IC
I/O IC
CIOS IC
PLS
SID
SCE
Guidance and Navigation Simulator
I O S
Univac 1100/44 Host Computer
Univac 1100/44 Host Computer
SWITCH
VIS IC
SID IC
I/O IC
CIOS IC
PLS
SID
SCE
Visual System
Motion Base
I O S
8/32
8/32
8/32
Network Simulation System
MCC
I O S
VIS IC
SID IC
I/O IC
CIOS IC
PLS
SID
SCE
Visual System
Fixed Base
I O S
8/32
8/32
SCE
SpaceLab Simulator
I O S
▨ = Shared Memory

Figure 1
Shuttle Mission Training Facility
Former Configuration

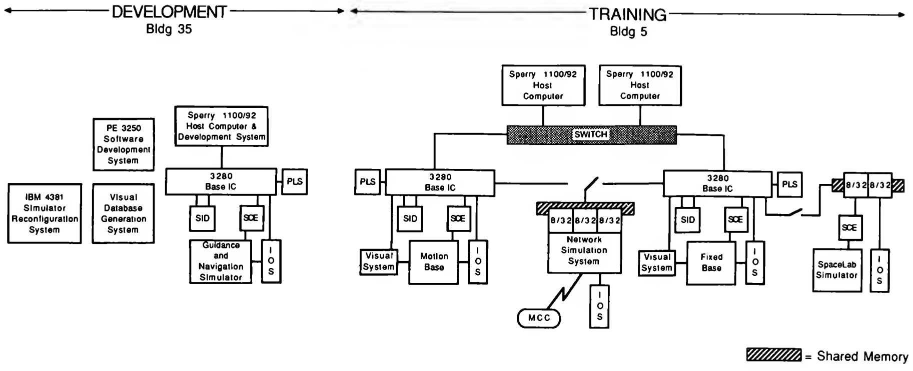

DEVELOPMENT
Bldg 35
TRAINING
Bldg 5
IBM 4381 Simulator Reconfiguration System
PE 3250 Software Development System
Visual Database Generation System
Sperry 1100/92 Host Computer & Development System
3280 Base IC
PLS
SID
SCE
Guidance and Navigation Simulator
I O S
Sperry 1100/92 Host Computer
Sperry 1100/92 Host Computer
SWITCH
PLS
3280 Base IC
SID
SCE
Visual System
Motion Base
I O S
8/32
8/32
8/32
Network Simulation System
MCC
I O S
3280 Base IC
PLS
SID
SCE
Visual System
Fixed Base
I O S
8/32
8/32
SCE
SpaceLab Simulator
I O S
▨ = Shared Memory

Figure 2
Shuttle Mission Training Facility
New Configuration

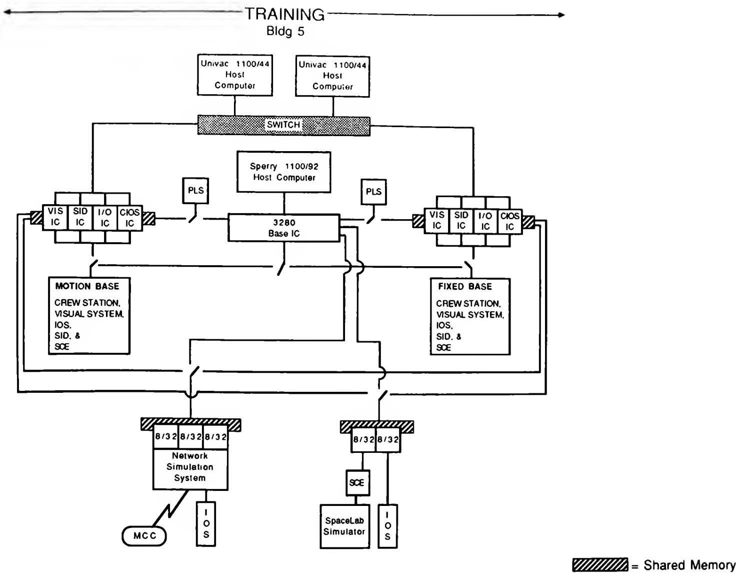

TRAINING
Bldg 5
Univac 1100/44 Host Computer
Univac 1100/44 Host Computer
SWITCH
Sperry 1100/92 Host Computer
PLS
PLS
3280 Base IC
VIS IC
SID IC
I/O IC
CIOS IC
VIS IC
SID IC
I/O IC
CIOS IC
MOTION BASE
CREW STATION, VISUAL SYSTEM, IOS, SID, & SCE
FIXED BASE
CREW STATION, VISUAL SYSTEM, IOS, SID, & SCE
8/32
8/32
8/32
Network Simulation System
MCC
I O S
8/32
8/32
SCE
SpaceLab Simulator
I O S
= Shared Memory

Figure 3
Shuttle Mission Simulator
Interim Configuration 1

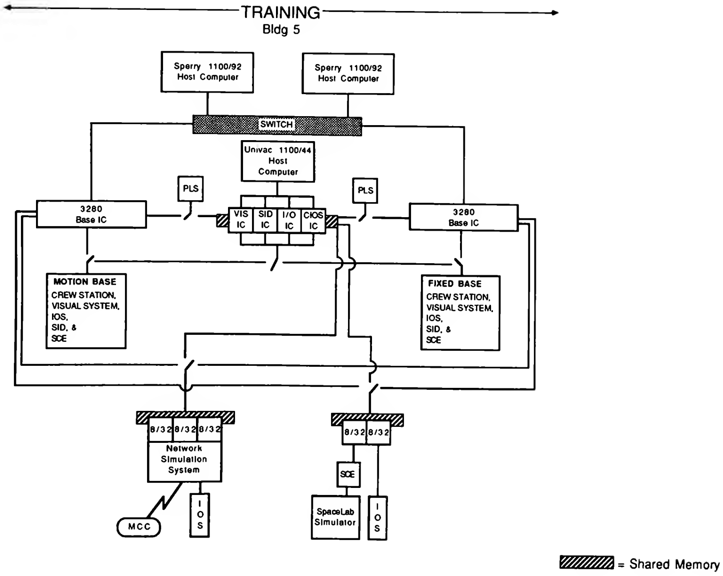

TRAINING
Bldg 5
Sperry 1100/92 Host Computer
Sperry 1100/92 Host Computer
SWITCH
Univac 1100/44 Host Computer
PLS
PLS
3280 Base IC
3280 Base IC
VIS IC
SID IC
I/O IC
CIOS IC
MOTION BASE
CREW STATION, VISUAL SYSTEM, IOS, SID, & SCE
FIXED BASE
CREW STATION, VISUAL SYSTEM, IOS, SID, & SCE
8/32
8/32
8/32
Network Simulation System
MCC
I O S
8/32
8/32
SCE
SpaceLab Simulator
I O S
▨ = Shared Memory

Figure 4
Shuttle Mission Simulator
Interim Configuration 2

A NEW CONCEPT IN REAL-TIME NETWORKING

John R. Jurgensen
Chief Engineer
SYSTRAN Corporation
Dayton, Ohio

Abstract

This paper discusses a new concept in networking that uses several unique features for high speed interprocessor data communication. These network features are summarized as follows: The network is data driven and, therefore, passes only those data that change, thus eliminating redundant data transmissions. It uses shared memory in that each network node has its own two megabytes of memory that are effectively made to look alike for all nodes. It has the ability to handle all commonly used CPU representations of integers, bytes, words, and long word formats in a manner transparent to the network user. It requires no real-time communication software, since all network communication is handled by the network hardware. Several other features are designed into the network that will be discussed in the body of this paper.

Nomenclature

- CSMA/CD - Carrier Sense Multiple Access/Collision Detection
- OSI - Open System Interconnect
- CSR - Command Status Register
- ECL - Emitter Coupled Logic

Background

SYSTRAN, under contract to Wright-Patterson AFB, developed a data driven network called SMARTNet 1A, utilizing a 20- Mhz basic clock frequency and 8K words of shared memory per node. The network was tested using six PDP-11/73s, one 11/40, and one 11/60 processor interfaced to an M372 F-16 Fire Control Computer and programmed with the F-16 six Degree-Of-Freedom equations of motion. Sensor models were assigned to the PDP-11/73s and the computing load was balanced around the network. The results of this demonstration were so successful that SYSTRAN produced an improved version called SMARTNet 1B, which used a clock frequency of 50 Mhz and 64K words of node memory. This was the final delivery version under SYSTRAN's contract to Wright-Patterson AFB. In an effort to develop an enhanced version of SMARTNet 1B, SYSTRAN initiated the development of a new product called Shared Common Random Access Memory (SCRAMNet)^{TM} Network. It has a much faster clock frequency of greater than 100 Mhz, a 30-fold memory increase to 2 Mbyte, a much improved interrupt structure, an enhanced communications protocol, and a much more modular architecture. The SCRAMNet Network will provide a very high speed collision-free communication link for small to medium real-time networks of less than 256 nodes. It will greatly reduce acquisition costs for simulation hardware and software and provide for easy addition or deletion of processors to match the computing load.

The following paragraphs will describe the design features embodied in the SCRAMNet Network.

SCRAMNet Network Overview

The main features of the SCRAMNet Network Node, summarized in Table 1, can best be explained in terms of operating features.

Table 1 - SCRAMNet Network Features

BANDWIDTH > 100 MHz

MEMORY--8K TO 2 MEGA BYTES

- DATA DRIVEN
- NETWORK REPLICATED

256 TOTAL NODES PER RING (NESTABLE)

- RING PROTOCOL
- UP TO 2000 FEET NODE SEPARATIONS

POWERFUL NODE INTERRUPT CAPABILITY

DIAGNOSTIC MODES

NODE LATENCY, 125 ns TO 660 ns

FULL MONITOR NODE CAPABILITY (OPTION)

SIMPLE TO USE

Network Shared Memory

The network is based on an innovation referred to as "shared memory". Shared memory is not to be confused with common memory. Each processor on the network has a network interface card with a block of 8Kbyte to 2Mbytes of network shared memory. The shared memory is physically part of the associated processor's memory address space. Each time the processor makes a "write" to its 2Mbytes of shared memory, that "write" is interpreted by the network logic as a request to transmit that memory cell around the network ring to each of the other network nodes. The token is composed of 83 bits and consists of the memory offset address, 32 bits of data, parity bits, and start/stop bits. As the data are passed around the network, they are written into each node's memory block at the respective memory offset address. Thus, each network node maintains identical copies of a 2Mbyte block of memory.

Data Driven

Greatly reducing the total network loading is the "data driven" feature. As each processor makes a "write" into its network memory, the network logic compares the datum being stored to the current memory contents to see if the datum has changed. If there is no change to the datum (as is frequently the case) the network Is not accessed. However, If a change Is detected, the next available token Is then used to pass the changed datum around the network rlng to each of the other nodes. Since tests on the F-16AB Block-15S Simulator have shown that typically 75% of simulation parameters remain unchanged from each 20 ms mlnor frame to another, this Increases the network effective bandwidth by a factor of four. In contrast, a typical packet switching CSMA/CD protocol (I.e., Ethernet and others) passes all data around the network whether changed or unchanged.

Rlng Protocol

The network uses a protocol which eliminates all problems of bus access contentlons. Messages are passed between nodes at a bit clock-rate of 100Mhz. Each node has a latency maximum of 0.6 micro seconds to pass the token, store the data, and then to re-transmit the data. Up to 256 nodes can be mounted on the network, and distances between nodes of 2000 feet can be realized using fiber optic transmission media. Should the network ever reach bandwidth saturation, It would continue to pass data at the bandwidth clock-rate until the network backlog is satisfied. However, typical applications Involving real-time simulation seldom load the network to more than two percent of bandwidth.

Software Requirements

Another feature of the SCRAMNet Network is that no software is required In any of the processors to pass data around the network. Apart from the software used to Initialize the network, network communication is performed by the network hardware in a fashlon transparent to the network users.

Modular Design

A significant facility cost savings can be realized using the SCRAMNet Network. The number of computers and network nodes can be easily expanded or contracted to fit the computing load requirements. It is a very modular approach to simulation, since little system software reconfiguration is required when a processor is added or deleted. Also, much smaller processors can be used to support large applications, since they can effectively run In parallel. Models can be distributed around the network to balance the computing load, and single function computers can be programmed to be, for example, an engine, an enemy aircraft, an air data computer, an inertial navigation system, etc. This generic and modular design also reduces software maintenance costs.

Network Topology

Figure 1 shows the typical topology of a SCRAMNet Network. Each computer has Its own SCRAMNet card. As a node receives the message slot In this network, It processes It and then transfers It on to the next node. This Is done simultaneously with all processors on the network, so that with 12 nodes (for example) offering data, there can be 12 messages transferred from one port to the next simultaneously. This allows for a very high throughput of data flow, because there Is very little wasted time between each of the respective processor nodes.

Network Monitoring System

A special Newtork Monitor System Is also available to monitor and record all significant network events by using special triggers and processor Interrupts. Although not required to make the network function, a larger simulation network would typically have at least one monitor node. The monitoring node consists of a stand-alone computer which Is fully programmed to:

- Record time-tagged network traffic
- Trigger on network events
- Capture data blocks on event
- Determine last node to write
- Measure network loading and status
- Display user selected variables
- Track errors in simulation
- Log all network traffic
- Generate network statistics

Interrupts

The SCRAMNet Network design also uses a sophisticated real-time interrupt scheme. The interrupts are set up during initialization using a one-byte extension which is appended to each SCRAMNet Network memory cell. This gives a total of 40 bits of memory per memory cell, 32 of which can be transmitted on the network and 8 (least significant) of which are needed to set up and control the interrupt structure for that word. (This potentially gives 2 million unique Interrupts at any SCRAMNet Network Node). The user initializes each interrupt using a CSR with Its interrupt Initialize bit set.

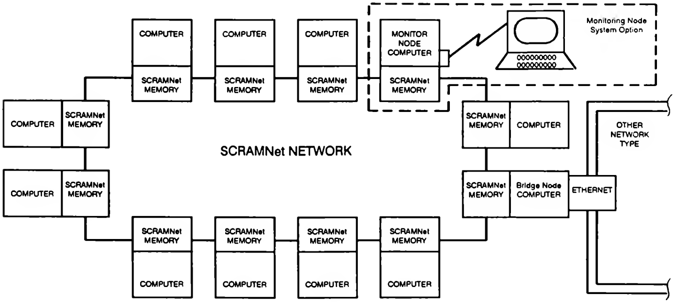


FIGURE 1 - SCRAMNet Network Topology and Node Types

There are four bits that can be controlled by the host, namely:

- RIE - Receive Interrupt Enable - which will interrupt the host processor any time data are written into that memory cell by the network.
- TIE - Transmit Interrupt Enable - which triggers a network interrupt message any time the host writes into that memory cell.
- ET1 - External Trigger 1 - causes a 30 ns pulse to an external connector pin when a write to this memory cell occurs.
- ET2 - External Trigger 2 - causes a 30 ns pulse to an external connector pin when a write to this memory cell occurs.

The ET1 and ET2 external interrupts give the user a powerful way of linking the network to the control of real-time events such as the timing of models and other real-time events, such as the control of hot bench avionics or other electronic devices.

All interrupt initialization is accomplished by each individual node and, once initiated, requires no further software control.

Programmable Byte Swapper

This SCRAMNet Network feature is truly unique in its function. Due to the differences between processors in the area of integer data representations, a problem could occur if, for example, a Motorola processor were to place integer data onto the network which had a DEC processor on it.

There are two major integer data formats describing how a processor associates the address of a byte with its significance in the data type in which it is contained; 1) Little Endian and 2) Big Endian. Little Endian is the integer format where the least significant byte of data is stored at a least significant address. In the Big Endian integer format, the most significant byte of data is stored at the least significant address. DEC and Intel both manufacture processors and mini/micro systems which use the Little Endian integer format, whereas Motorola manufactures processors and systems using the Big Endian format.

The SCRAMNet Network provides a hardware reformatting solution to this problem. It is implemented using two bits in the CSR2 register and functions as shown in Figure 2.

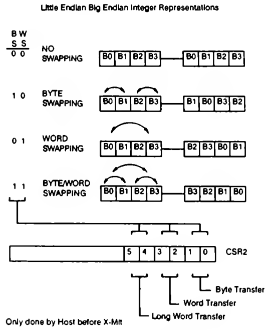


Figure 2 - Programmable Byte Swapping

Traditional Network Problems and Solutions

Up to this point the basic design of the SCRAMNet Network and its features has been discussed. Now a discussion of the traditional network problems will be given along with a discussion on how the SCRAMNet Network handles these problems.

The problems typically found in a network are:

- Network Response Delays
  - bus access delays
  - operating system delays
- Throughput Bottlenecks
- Error Detection and Recovery

Network Response Delays

Network response delays can occur in several areas. The most critical areas concern the operating system software delays in processing the data in and out of the computer, the delays in waiting for bus access, and the actual transmission delays once access is granted. Commercial networks typically quote the clock-rate throughput for this data transfer and tend to ignore what happens as network traffic increases and access collisions occur. Even if no collisions occur, what is the software overhead associated with processing the data in and out of the network buffers? Ultimately the network performance should be judged by how long it takes from the time an application program writes to its memory until these data are read by the users of that data. In simulation, or any other real-time application, the "Application-to-Application" delay is the single most important quality of a network. This quality will be addressed in its two areas in the following paragraphs.

Bus Access Delays. The bus access is by far the most significant source of network delays. It is directly related to the number of nodes and the amount of offered traffic per node. Access delays can amount to 0.25 to 1.25 seconds. In the case of a heavily accessed CSMA/CD packet-switching network delay can cause a total system crash. For on-line users this is a minor annoyance, but for real-time applications it can be devastating. Token rings have only minor problems with bus access, but most token ring networks only have one token circling the network at any given time, no matter how many nodes are present on the network.

The SCRAMNet Network, on the other hand, uses a modified token ring with one token per node. The worst case delay is only one token slot for any bus access by any node. This amounts to less than 0.6 micro seconds.

In Figure 3, the typical LAN access is shown by protocol. As throughput increases, the CSMA/CD networks rapidly drop off to an intolerable delay. Token rings have higher overhead (and therefore lower throughput) at the lower offered traffic levels, but rapidly pass up the CSMA/CD networks as the traffic increases. The SCRAMNet Network is estimated to lie somewhere between the typical token ring and the perfect network. (See figure).

Operating System Delays. Operating system delays generally are concerned with time required to pack and unpack all data blocks, to queue up the network hardware for a network transfer, and to service the network interrupts. These delays are related with seven layer OSI model and the software used to process data through each of these layers as it goes from the APPLICATION, PRESENTATION, SESSION, TRANSPORT, and NETWORK layers. To illustrate how much overhead can be added, the example is used of a one-byte transmission from one computer to another (Figure 4). Up to 256 bytes of overhead can be added to the one byte of data as it propagates down through the OSI software layers to the physical network. Then the data must wait for an average of 8 ms to gain access to the network

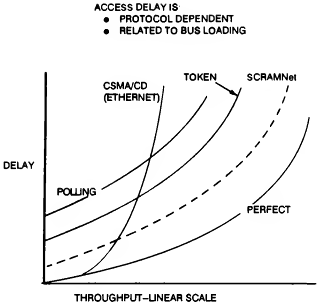


FIGURE 3 - LAN ACCESS DELAY BY PROTOCOL

media. If no collisions occur, the actual hardware transmission can be completed in about 0.256 ms. Then the target computer must strip off the 256 byte address information as it propagates back up to the using application program. This total overhead can be significant.

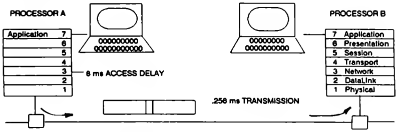


FIGURE 4 - OSI Interaction Example

The SCRAMNet Network handles this problem by using no network software in the transmitting or receiving node. As data are written to memory and changed from the last write to that memory cell, a token is initiated which passes the data around the network to each of the other nodes where the data are immediately stored. If data are time-skew critical, an interrupt can be set to occur on the last data element in a block to cause the receiving program not to use that data until the block of data is complete.

At the end of a simulation frame most networks transfer data in blocks, as shown in Figure 5. With the SCRAMNet Network the data elements are transmitted immediately upon a write to memory with virtually no delay. Up to 1.48 million, 32-bit data words can be transferred in one second for a 10-node network. (A 10-node SCRAMNet network could transfer 29,600 32-bit data words 50 times per second, for example.)

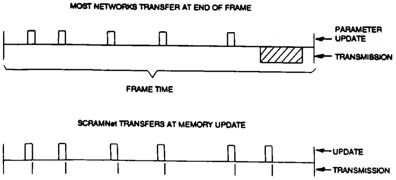

MOST NETWORKS TRANSFER AT END OF FRAME
PARAMETER UPDATE
TRANSMISSION
FRAME TIME
SCRAMNet TRANSFERS AT MEMORY UPDATE
UPDATE
TRANSMISSION

FIGURE 5 - SCRAMNet NETWORK DATA TRANSFERS EXAMPLE

Throughput Bottlenecks

Another network bottleneck to high throughput is the transmission of redundant data. This bottleneck is found in all networks. The problem occurs when data blocks are periodically scheduled for transmission at some point in a real-time frame. The entire block is sent, whether it changed or not. To filter the data in software is next to impossible because:

1. The data elements must remain contiguous to map properly into the receiving computer's database.
2. If unchanged data were to be deleted, address data would have to be appended to inform the receiving computer as to what information it was receiving.
3. Software filtering would add significantly to the overhead associated with a data transmission.

Typical simulations or real-time applications only change a measured 25 percent of the data in a typical minor frame. Therefore, networks are scheduling four times more data onto the bus than is needed. This factor alone could greatly relieve a network of traffic and increase throughput.

The SCRAMNet Network solves this problem by using a hardware filter to compare writes to network memory with what is already in memory. Redundant writes are allowed to take place, but no network access is initiated. This frees up time and token slots for other high priority traffic and goes a long way toward reducing network bottlenecks.

Typical throughput by protocol is shown in Figure 6. With the hypothetical "perfect network," as the offered traffic increases, the throughput starts to taper off until it hits the limit of the bandwidth or the transmission rate of the network itself.

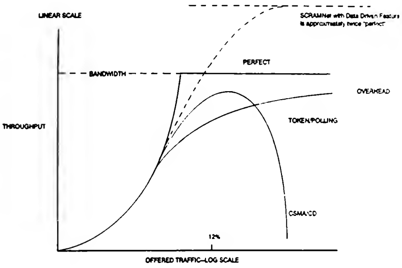


Figure 6 - Throughput-By Protocol

In a CSMA/CD network when transmissions approach 12 percent of the total bandwidth of the network and if there is a sufficient number of nodes, there will be collisions on the network that can sometimes require seconds to recover from. When the user is working on a terminal this is seen as a delay on the update of the screen. On the other hand, with the token polling approach using the SCRAMNet Newtork, as the offered traffic increases, the increase in throughput starts to taper off until it reaches some straight line that is about two times higher than a bandwidth limited "perfect network". This is due to the data filtering which so greatly increases the effective bandwidth. The typical token ring network approaches a straight line that is lower than the higher bandwidth line due to the "overhead" number of bits associated with the token. It should be pointed out that on a CSMA/CD network there are traffic conditions at which you can get higher throughput than with a token polling network of the same bit frequency.

Error Detection and Recovery

There are many ways in which a network can experience errors. The real issue is not how many transmission errors occur, but rather how many occur without being detected. All networks can usually recover from an error, but not all errors are detectable. Basically, the network errors fall into the following classes:

- Bit dropouts or insertions
- Node failures
- Bit state change

Bit dropouts or insertions

This error is virtually nonexistent, but if it does occur, it will result in framing errors in the SCRAMNet Network, causing a retransmission of the errored token by the originating node. Retransmission occurs in approximately one Loop time.

Node Failures

A node failure cannot be easily detected by the host computer. If undetected, node failure for the SCRAMNet Network effectively becomes an open circuit. There is an option available that will provide an optical by-pass upon power failure. This effectively acts as though the node disappeared, but the media integrity was maintained. If the host can detect its own node failure and if the failure is not in the immediate receiver/transmitter logic a bit can be set in the CSR which will allow an automatic by-pass at the ECL level.

Bit State Changes

This error can affect the network in many different ways depending on which message field is experiencing a bit change. If the address field in the token is changed, the address could be a nonexistent address. Normally, the messages circulates the ring until it is again received by the originating node, at which time it is cleared from the ring. The SCRAMNet Network handles the nonexisting address by incrementing an age counter at each node until it reaches 256, at which time it is cleared from the ring. If there are 64 nodes on a ring, for example, the message would circle the ring four times. Meanwhile, the initiating node will time-out after its address fails to circle the ring, and will reinitiate the message in slightly more than one ring time. This example assumes that an even number of bits in the address changed and therefore didn't get caught by the built-in parity checks.

If a data field in a message is changed, it will be caught by the initiating node when it completes the first loop circuit. A retransmission of the errored node will automatically occur. No other node will use the data if it fails parity checks.

If a control bit in the message is changed, it will fail parity and will be rejected by each node. Retransmission will be initiated by the originating node.

If the offset changes by an odd number of bits, it would fail parity and no attempt would be made to store the data. However, if the offset changes by an even number of bits, a valid datum will be stored in the wrong address and no recovery is possible.

The Langley Advanced Real-Time Simulation (ARTS) System

88-4595-CP

Status Report

Daniel J. Crawford,
Jeff I. Cleveland II and Richard O. Staib
NASA Langley Research Center
Hampton, VA 23665-5225

The real-time flight simulation system at Langley has been replaced with a modern system of digital networks. This system of ten Computer Automated Measurement and Control *(CAMAC)* ring networks supports data communication among the 15 active flight simulation sites and the 2 simulation mainframe computers. Three major innovations were developed for *CAMAC* to make it suitable for flight simulation: A block transfer capability to increase data rates to 3 bytes per microsecond, fiber optic converters to increase distance between sites to more than 3 kilometers, and a network switch to provide the instant creation of a network from any arbitrary combination of simulation sites. This system (ARTS) was installed in April 1987. By early 1988 all simulators were converted and the old system was removed. This paper is intended as a status report on the ARTS system. It briefly describes the architecture and principal subsystems including: The *CAMAC* network system (hardware and software), the clocking system, the signal converters, the control consoles, and the minicomputer and microcomputer interfaces. The performance and reliability of the system exceeds expectations and component failure data over an 11-month period are presented. Planned enhancements, including the replacement of the mainframe computers, are discussed.

INTRODUCTION

The Advanced Real-Time Simulation (ARTS) System supports flight simulation at the NASA Langley Research Center. It is composed mainly of: Data communication equipment, signal converters, an interval-timing subsystem, consoles to control and monitor individual simulations, local site computers, and a large amount of supporting software. At present, two mainframe scientific computers are used for computing flight system models and these have been integrated with the ARTS system. The system is centered around multiple Computer Automated Measurement and Control (CAMAC) networks, normally one per job, and a network-configuration switch which allows networks to be defined dynamically at job start-up time. Work began on the ARTS system in 1984 and the system was released to the users in mid-1987. At this writing (mid-1988), the system has been operational for one year. The system was described in detail in reference 1 in late 1986. Reference 2 is a shorter version of the same paper but it has a wider distribution and is consequently easier to acquire. For the sake of completeness, a short description of the system is included here but the main purpose of this paper is to update the earlier papers and to relay experiences and evaluations of the system since being released to users. Planned additions and enhancements to the system are also discussed.

The flight simulation facility at Langley is used to support research and development of technology in areas such as automated and augmented control, handling qualities, guidance, navigation, flight management, air traffic control, air combat, and workload analysis for pilots and astronauts. In general, the facility is not used for crew training. During a typical month, as many as 15 of these studies are in an active state. Each study is assigned two or three 3-hour periods per week to conduct research simulations. Higher priority jobs are assigned more time. During their run-time, a team of engineers is assigned a console, one or more cockpits, and the necessary display system support, both instrument and out-the-window. Pilots (test, military, and commercial) man the cockpit(s), especially during the data-taking phases of the study. In air traffic control studies, more piloted simulators are involved and both actual aircraft and simulators from other U. S. research facilities are occasionally integrated into the study.

The operational environment at Langley is quite different from airline facilities, aircraft manufacturing facilities, or military training facilities. These other facilities dedicate a significant portion of their resources to a particular vehicle or study for long periods of time (years). Such resources include cockpits, consoles, and in some cases computers and display systems. These differences in mission (general flight research versus specific vehicle research, or training) determine some of the requirements on the simulation system. For example, the network configuration switch discussed in this paper has value primarily when different complements of equipment are required to support different studies during a typical day. This sharing of equipment among studies costs less than the alternative, but generally, gives the pilot a somewhat less realistic environment in which to work. At Langley, the pilot is flying in a general-purpose fighter cockpit or general-purpose transport cockpit rather than a truly realistic vehicle-specific cockpit. Test pilots accept this general-purpose implementation better than operational pilots, but the negative effects seem to be minimal. The Langley cockpits are continually upgraded to reflect developments in the industry and to include study-specific equipment.

Other requirements which drive the design of simulation systems are more general and include a powerful scalar computing capability and a high-speed data communication system with minimum latency times when compared with frame times. The frame time is usually equivalent to the numerical integration step size used in solving the system model. Two factors dictate the requirement for processing power. The first is the complexity of the model and the second is the sample rate required by the model. Both of these factors have been growing over time as more responsive aircraft with structural flexibility and more complex control and weapons systems are being designed. Simulation at Langley requires no less than 20 steps per cycle with a one-pass integrator, normally Adams-Bashforth. Limitations on compute power force acceptance of this relatively coarse implementation. At present Langley has the capability of using frame times as low as 5 milliseconds when the model complexity and available computing power allow for solution in that short a period. Two CYBER 175 computers are available to solve the equations of motion and can be tightly coupled through shared extended memory to work on a single large simulation. More typically, one of these computers handles a single simulation or two simulations simultaneously. Typical frame times are 20, 25, or 40 milliseconds and it is not uncommon for three or sometimes four independent simulations to run on the system at the same time. Each of these computers has the power of approximately 10 VAX-MIPS or scalar LINPAK of 2.1 megaflops at 60 bits of precision. An effort is underway to increase computing power by replacing the real-time computers.

Latency times are usually caused by excessive involvement of a slow operating system, by transport delays, and buffering blocks of data during input and output operations. The first effect of latency is to increase input/output (I/O) time, thus decreasing the time available to compute the model. This could force (because of limited computer power) an undesired simplification of the model or the use of a longer frame time than is prudent. The second effect, which is more extreme, occurs when latency times cause additional frame delays, i.e., when data passing from one part of the system to another is delayed by more than a frame. In this case, extraneous roots (eigenvalues) are introduced and large errors compromise the integrity of the simulation. This is usually evidenced by less stability in the simulated system than in the original system.

The ARTS system was designed to be reliable and modular and to be flexible in configuration. In addition, the system was designed to interface the Langley host computers, site computers, and cockpits, and to introduce minimum latency times in I/O operations. All software, the clock subsystem, the switch controller, diagnostic equipment, the control consoles and site computer interfaces were designed and built in-house. Almost all of the CAMAC equipment (networks, switch matrix, signal conversion equipment, site microcomputers, etc.), was purchased from a single vendor (Kinetic Systems Corporation) and is available now as catalog items. The system is maintained by an on-site support contractor. Three to six people (engineers, technicians, and analysts) are assigned for maintenance and development of the ARTS system. In addition, each simulator has at least one technician assigned. Teams of NASA engineers normally design and conduct the research studies with the assistance of simulation engineers, analysts, and programmers who are both NASA personnel and support contractors.

The Langley simulation facility has been in operation for over 30 years. Early work included the simulation of rocket aircraft, rocket boosters, spacecraft, and conventional aircraft. During the sixties much work was done on rendezvous and docking of the Gemini/Agena pair and of the Apollo missions. There is heavy emphasis on automatic control and control augmentation of flight vehicles. In current times, no modern aircraft is developed without a corresponding parallel vehicle simulation and many of these are being tested and modified by Langley engineers in this simulation facility.

SYSTEM DESCRIPTION

Overview

The ARTS System has five major components. The first is the system of CAMAC networks. These networks are driven by two CDC CYBER 175 computers coupled to the configuration switch to allow dynamic reconfiguration of the networks. The second is the site CAMAC crates which contain the simulator interface equipment such as signal converters, both digital-analog and digital-digital. The third component is the interval timing subsystem. The control consoles for simulation control and monitoring constitutes the fourth component, and the fifth is the central and distributed software.

Figure 1 is a schematic representation of a single network which has been configured to include three sites: A control console, a 6-degree-of-freedom motion cab, and an out-the-window display generator. The network is a CAMAC enhanced byte-serial highway ring with the master (Serial Highway Driver) integrated with the CYBER channel. The CYBER channel is 12 bits wide and has a 500-nanosecond clock period. The maximum channel data rate is 24 megabits/second and the Dataway (crate backplane) also has the same maximum data rate. The Dataway is 24 bits wide and has a period of one microsecond. The user (or net) block data rate for the system is 24 megabits/second. The CAMAC network is 8 bits wide plus a clock signal. The configuration switch is capable of handling up to 12 highways and up to 44 sites. At each of the site/highway crosspoints there are eight switched signals. Each highway is a ring network starting from the Serial Highway Driver, passing through the switch and returning to the Serial Highway Driver. When a site is selected, the ring is diverted to the site; when the signals are returned from the site they are reinserted into the ring. A site which is not included in a network is said to be bypassed. A site can be selected by only one highway, which prevents network-to-network collisions and interference.

The configuration switch has a microcomputer controller which is labeled "switch control" in figure 1. When a new job enters the system (as an interactive job on the CYBER), the job requests a group of specific sites. The job scheduler on the CYBER sends this request via RS232 to the switch controller. The controller checks the site tables it maintains for site availability. If available, the requested sites are reserved for the new job. If not available, the job is not allowed to enter real-time status. These switch controller tables provide status of the site assignments and active jobs. This status is displayed on television monitors at various locations throughout the simulation facility. In addition, the status can also be displayed on any CYBER terminal by invoking a control statement. When new simulation host computers are added to the system, they will be integrated with the switch in the same manner as the CYBER computers shown in figure 1. It should be noted that almost any computer which provides an interface to CAMAC can be integrated into the ARTS System.

The computers, the highway drivers, and the configuration switch are centrally co-located. The sites reside at distances between 350 feet and 6000 feet from this central location. The fiber optic U-ports (modems) shown in figure 1 are necessary to drive the wide bandwidth (50 megabits/second) bit-serial signals over these long distances and provide electrical isolation. In a rack adjacent to the configuration switch, the 8-bit parallel electrical signal is converted to a single-bit serial optical signal and transmitted to the site. At the site the signal is converted back to an electrical signal and passes through all site crates (usually 2, 3, or 4 crates). Leaving the site crates, the signal is again converted to optical and sent back to the central location where another conversion to an electrical signal occurs and the signal reenters the switch. Presently there are 25 sites cabled into the switch. Seven of these are inactive. Of the four fibers in each cable, two are used for the network, one for the timing subsystem, and one is a spare. This star-like pattern with the switch at the center allows configuring ring networks from arbitrary groups of sites. The disadvantage or peculiarity of the star pattern occurs when two adjacent sites are connected into the same network. In this case, the signal is sent out to one, back to the switch, out to the second site and then back again to the switch. The worst-case travel time associated with a round trip to a site is less than 20 microseconds and therefore can be tolerated. The configurability afforded by the switch meets an important basic requirement of simulation at Langley.

Table 1 represents an attempt to quantify the system. Twenty-five fiber optic cables have been installed. The configuration switch has boards to accommodate 20 sites with 16 operational. The table shows the various types of sites--both operational and planned. The four operational laboratories are the crew station system research laboratory, the air traffic control facility, the EASILY B737 hot bench facility, and the Advanced Concepts Simulator (ACS). Each of the 16 operational sites has local intelligence (either a minicomputer or a DEC LSI 11/73 microcomputer). Some of these microcomputers use a PC-XT clone in lieu of a DEC VT220 terminal and thereby have a hard disk available and some software for site statistics gathering. It can be seen from the table that Langley intends to increase the number of minicomputers in the system. Five of the planned new minicomputer sites involve new graphic display capabilities. Another growing requirement which is not shown in table 1 is the ability to communicate through the site crates to other data buses such as RS232, ARINC, MULTIBUS, VME, Mil Std 1553, and IEEE 488. Although the requirement for digital-to-digital communication was anticipated, its magnitude and immediacy was underestimated.

Why CAMAC?

There are several ANSI/IEEE and international standards defining the Computer Automated Measurement and Control Network (see reference 3). CAMAC was developed by physicists and engineers principally for data acquisition and control of nuclear particle accelerators. It has world wide acceptance in that field but is not otherwise well known. In the early 1980's while searching for a network to replace the analog signal distribution system, it became apparent that modern networks would have difficulty meeting the requirements of real time simulation. The high data rate requirement meant that there were only a few candidates for selection and that the equipment would be expensive. The more difficult problem was latency time caused by block buffers at the network interfaces. To illustrate this effect, first consider sending an 1800 byte block of data from point A to point B over a 24 megabits/second line. This would take 600 microseconds plus a few microseconds for transport delay. Now if an interface unit with a block buffer is used at both point A and point B, travel time from A to B would increase to 1800 (+) microseconds. This is tolerable for file transfers but would seriously restrict a real-time simulation running at a short frame time such as 5 milliseconds. Some networks compound this problem by using two buffers in each interface unit and additional time is lost as data travels from one buffer to the other. The CAMAC network does not buffer blocks of data. The data travels from the CYBER channel through the serial highway driver, across the network to the addressed crate, through the serial crate controller to the crate backplane (Dataway), and arrives at the memory of the addressed module (RAM or registers). The CAMAC network acts like a very long 24 megabits/second computer channel and this is one of the main reasons it was selected. In addition, the performance of this network with a single master, is predictable, and worst-case data transfer times can be predetermined. This predictability is a requirement for real time job scheduling that cannot be met by probabilistic type networks, such as ETHERNET. Token passing networks which might possibly be applied to simulation were not available when CAMAC was selected.

Another system requirement that CAMAC met was connectability. With CAMAC, a network can be quickly configured from arbitrary groups of sites in support of particular research studies. In the past a system of patchboards was used for connectability on the analog signal distribution system. For the CAMAC network, the network configuration switch was developed and placed into service. For other networks, a practical response to the connectability requirement is not obvious. The CAMAC network was also selected because of its modularity, generality, and because it is built to a recognized standard. These three qualities, cited with regularity in the justification phase of the project, have been very significant in the development and subsequent operation of the system.

CAMAC Enhancements

There were, however, three capabilities in CAMAC that needed to be enhanced. The configuration switch needed to be developed in order to gain the site connectability discussed above. The maximum allowable distance between sites needed to be extended and the effective data transfer rate needed to be improved.

Kinetic Systems Corporation (KSC) developed the configuration switch matrix to meet the connectability requirement. The matrix is housed in a modified Fastbus chassis containing an input card, an output card, and, at present, five pairs of site cards. Each pair of site cards services four sites. The configuration switch controller, interface unit, and video status display equipment were built in-house by

NASA. Each of the 12 possible highways passes through all the site cards and each site is switched in or bypassed. Secure jobs are not run through the switch; instead, these jobs use a private serial highway driver. The appropriate sites are manually disconnected from the switch and cabled directly into a ring network.

Initially, the longest distance expected between sites was less than a 1,000 feet, the limit that can be expected from high-frequency electrical signals. Because of this limitation, the plans included only two adjacent buildings in the high-performance networks. Lower performance alternatives were considered to reach outlying sites. However, KSC removed the requirement for this alternative by first developing a 1-kilometer, 50-megabit/second fiber optic U-port and then shortly thereafter developing a 3-kilometer unit. Both of these units use light-emitting diodes. This permits connecting sites within a 3-kilometer radius of the configuration switch. KSC is now developing a 20-kilometer unit that uses a laser, allowing distant sites to be considered for inclusion in the system.

In order to meet the stringent data rate requirements, KSC developed a true block data transfer capability for the CAMAC highway. Prior to this development, a message contained no more than 3 bytes of data and, with protocol, would be up to 10 bytes long. Previously, a block was defined as a group of adjacent messages. The overhead in this scheme was too high to achieve the rates required. The newly defined block is transmitted as a message of arbitrary length packed 3 data bytes per 5 bytes on the network. The block is clocked at the CAMAC standard 5 bytes/microsecond. There are additional overhead bytes at the beginning and end of the block. R. T. Cleary discusses the block transfer capability in reference 4 and shows that it asymptotically approaches a 24-megabit/second effective data rate as message size increases. This rate is three to four times faster than the standard CAMAC data rate.

Three major system components had to be developed for the block data capability. The Block Transfer Serial Highway Driver (BTSHD) is the network master and interfaces to the CYBER channel. The Block Transfer Serial Crate Controller (BTSCC) is the crate interface to the network and is the Dataway bus master. It resides in the rightmost two slots in each crate in the system. The List Sequencer Module (LSM), on writing, allows splitting a block of data among several target modules in a crate and, on reading, allows gathering a block of data from several source modules in a crate. The LSM does not buffer data or slow the data transfer rate. There are two block transfer modes. The first is the transfer to or from a single module such as a FIFO or a RAM. In this mode the BTSCC repeats the Dataway bus address and function for each 24-bit data word in the block. If necessary, the module will manage incrementing its memory address. In the second mode, the Dataway bus addresses and functions are preloaded in the LSM memory at job start-up time for a particular block. An LSM can accommodate two read lists and two write lists. Each list can have as many as 512 entries. During the block transfer, these addresses and function bits are fetched from LSM memory and placed onto the appropriate bus lines simultaneously with the 24-bit data word. The BTSHD and the BTSCC were built not only to accommodate these block modes but also to process the older standard short message modes.

Site Computers and Control Consoles

Each site consists of one or more crates connected to a serial highway and the hardware associated with that site, such as cockpits, control consoles, and graphics generators. The typical complement of site modules is listed in table 2 and table 3. These tables also contain the total number of modules in the system. The Minicomputer Interface Module (UMIM) and Microcomputer Interface Module (QMIM) were developed in-house and serve as speed and protocol matching devices between the computer and the crate Dataway. A typical site microcomputer configuration using the QMIM is shown in figure 2.

The site computers (LSI or attached micro or mini) serve three purposes. First, the site computers provide local intelligence for off-line quality assurance and diagnostic testing. The site computers are used to test the entire crate system including the backplane bus, attached conversion equipment modules, and attached simulation cockpits and hardware. Second, the site computers are used for off-line demonstrations. In this function, the computer may play back previously recorded simulation data or may compute, in real-time, the response of a simplified simulation model. Third, the site computers are used for both control of external devices and local communication during actual flight simulation. Site computers control RS232 communications with cockpit devices since a compatible RS232 CAMAC module is not yet available. Site computers can be used to interface with General Purpose Interface Bus (GPIB) modules in the crate to remove a complicated protocol burden from the host computers.

A special site used by each simulation application is a control console. There are presently four of these consoles used by the simulation programmers and the research engineers for control and monitoring of the simulation activities. A photograph of a console is shown in figure 3. The console contains two CAMAC crates that house an LSI 11/73, conversion equipment, an SCIU used for converter triggering and host computer clocking, and a QMIM. The console contains a Jupiter 12 color graphics system with an Elographics touch sensitive screen overlay. The Jupiter 12 with its 68000 processor is used as the main control element for simulation. The screen contains a display of switch buttons and the touch panel provides the switch inputs. Communication with the host computer is coordinated from the Jupiter 12 via direct memory access to the console crate LSI 11/73 using a custom protocol to maintain high speed. The console also contains the turret assembly which contains a variety of analog and digital devices including a cockpit 2-axis hand controller, slide potentiometers, dial potentiometers, alphanumeric displays, annunciators, etc. Digital devices in the turret assembly are controlled through a Prolog microprocessor via RS232 communications with the LSI 11/73. The console was designed and built in-house and has been well received by the simulation programmers and research engineers.

Clock Subsystem

The purpose of the clock subsystem is to provide the simulation frame rate and to synchronize simulations. The clock subsystem is composed of a clock central unit and multiple Site Clock Interface Units (SCIU). These SCIUs are CAMAC modules connected to the central unit by means of a separate fiber optic star network. A block diagram of the clock subsystem is presented in figure 4. Two distinct time signals are broadcast by the central unit to each site on a single fiber. The first time signal, called the frame tick, has a constant 500-microsecond period. A frame tick count is set in a register in the SCIU by scheduling software to establish a frame time. The frame tick count is decremented once for each occurrence of the frame tick. When the tick count reaches zero, each SCIU for the simulation issues a job-specific periodic signal called frame compare which is the beginning of a real-time frame. The frame time is determined independently for each job but must be a multiple of 500 microseconds. The second time signal, called the job sync tick, has a longer period called the Clock Common Multiple (CCM). The CCM is set by thumbwheel switches on the clock central unit to a multiple of 500 microseconds from 1 to 65,535. This longer interval is used to synchronize jobs. A requested frame time must divide without remainder into the CCM. The SCIU will wait to issue the first frame compare of a session until a job sync tick occurs. These two constraints guarantee that all jobs in the system, regardless of frame time, receive a frame compare simultaneously with each occurrence of the job sync tick.

The clock central unit uses an accurate temperature-controlled 5-megahertz oscillator from which it derives the two time interval signals. It combines the two signals into a single 1-megahertz signal and uses fiber optic transmitters to send the signal out to the sites. The central unit contains an Intel 80/24 single board computer and an Intel 534 communication board for communication with the mainframe processors. Each site contains an SCIU which is a CAMAC module residing in one of the site crates. Each SCIU has a fiber optic receiver that decodes the two time intervals from the central unit. The SCIU frame tick count register is set by scheduling software at job initiation time. At all but one site for a simulation, the only function of the SCIU is to count down the central unit ticks and when the count goes to zero, issue a trigger pulse to start the conversion process in all the relevant CAMAC modules. At one site for each simulation, usually the control console, the SCIU is preset for an additional function. On each frame compare the SCIU initiates a demand message which is sent up the serial highway to the mainframe where it starts the real-time frame process (input-compute-output). Figure 5 illustrates the input, compute, output frame process used for simulation at Langley. When the input/output peripheral processor (PP) on the mainframe receives the frame compare from the serial highway, the PP begins the input phase of the real-time frame. During this phase, the PP transfers all synchronous data from the site crates including analog-to-digital conversions, discrete inputs, and other input data to the mainframe central memory. Once input is complete, the PP interrupts the mainframe and the simulation program is then placed into execution. Once the equations of motion for the simulation have been computed for the current frame, the program signals the PP that synchronous output is ready. The PP transfers output from the mainframe central memory to the site crates where the data waits in the conversion module registers for the occurrence of frame compare, which will then cause the data to be converted. After output has been transferred, the PP waits for the next occurrence of frame compare to begin the cycle again.

Signal Conversion Equipment

Each crate may contain signal conversion equipment to convert site signals to CAMAC form. Two types of input converter modules and three types of output converter modules were designed and built to Langley specification in support of the ARTS project. These modules are Analog-to-Digital Converters (ADC), Discrete Inputs (DI), Digital-to-Analog Converters (DAC), Digital-to-Synchro Converters (DSC), and Discrete Outputs (DO). Table 3 summarizes signal conversion equipment usage. The ADC's, DAC's, and DSC's are 16-bit devices with 14 bits of accuracy. The data word transmitted uses 16 bits with 14 meaningful bits. This allows the precision of the units to be increased without major changes in protocol or software. The conversion can be triggered either by the SCIU or by a CAMAC function from the host computer or site computer. Normally, conversion is triggered by the SCIU. All three output module types are double buffered. These modules may be initialized to convert on the next clock compare or immediately upon arrival of new data. Normally, conversion is triggered by the next clock compare. The DO modules have one of three types of output: Isolated relay contact, optical isolation, or logic level. Experience has shown that isolated relay contact type of DO modules are best suited for use at Langley.

MINICOMPUTER NETWORK INTERFACE

As discussed earlier, the requirement to connect various minicomputers and data buses at the simulation sites is stronger and more immediate than anticipated. The system description section discussed the site microcomputer interface (QMIM) and the manner in which the LSI 11/73 was used to manage some low-speed data buses. A second module for computer interface called the UMIM was also developed in-house for communicating through the DEC DR11-W, direct memory access card, to the VAX Unibus. The UMIM is a much more general interface than expected since the DR11 interface has become a de facto standard and is emulated on several non-DEC machines. Table 4 lists some of the equipment that have been or will be interfaced. Two VAX and two PDP 11 computers have been connected to the network. Connection has been made to the ARINC bus on the EASILY B737 hot bench by means of a DR11 card, a processor card, and an ARINC card, all mounted in a VME chassis. It is reported that KSC is developing both an ARINC and a Mil Std 1553 CAMAC module to simplify these interfaces. Langley is acquiring a CT6 Computer-Generated Imaging (CGI) system from Evans and Sutherland (E&S), which uses a Gould Concept computer as the front end. E&S is working with Gould in Salt Lake City to check out their DR11 type interface with the Langley-developed UMIM. The CYBER/CGI interface is expected to be operational at Langley in the fall of 1988. As can be seen in the table, there are several other planned interfaces including replacements for Adage graphic generators and the target generators for three projection spheres. The effort for development of a minicomputer interface module for a computer without a DR11 capability is estimated at less than one man-year.

Figure 6 illustrates how the CYBER is connected to a site computer by means of the UMIM. The blocks labeled

"ACC," "Converters," and "UMIM" represent CAMAC modules which are inserted into the crate backplane. The signal converters are included to illustrate that they can coexist with a UMIM. For example, the MOTAS site contains two computer interfaces plus some Analog/Digital converters. The blocks labeled "Bus Interface," and "DR11-Type" are cards built for the site computer bus. The UMIM has two 2000 byte FIFO's; one to the site computer and one from the site computer. The UMIM uses a site computer interrupt in one direction and the CAMAC LAM (pseudo interrupt) in the other direction as part of the protocol. Significant latency time can be introduced in the communication between the site computer memory and the UMIM FIFO. Most of the delay can be attributed to the site computer operating system and can be minimized by customizing the DR11 driver and avoiding certain I/O service calls during synchronized real-time.

The "local path" which uses an Auxiliary Crate Controller (ACC) as shown in figure 6 allows the site computer to communicate with the site crate modules. The site computer can write or read the converters either when running or, as is more common, during daily site checkout. The ACC and Bus Interface card are purchased as off-the-shelf items. One such path per site crate is required. Most of these minicomputer sites have only one crate. If the equipment or software for the local path were not available, then a site microcomputer would be used to do off-line checkout. Most of these minicomputer sites require a Site Clock Interface Unit (SCIU) module which generates a periodic frame tick used by the converters and can be connected to a site computer interrupt. The SCIU was also built in-house and is discussed in the clock subsystem section. To summarize figure 6, the UMIM path is used strictly for host/site computer communications whereas the local path allows the site computer (in addition to the host) to read, write, and function local CAMAC modules.

DIAGNOSTIC CAPABILITIES AND QUALITY ASSURANCE

There are three major types of hardware problems that can possibly occur in the system. These are module problems, crate problems, and network problems. Module problems are easy to isolate. Technicians who manage the simulators usually know when a converter goes bad. Otherwise, the trouble is detected during the daily quality assurance tests on the equipment with the local LSI microcomputer. In either case, the module can be replaced in less than a minute. The module is then checked and when necessary sent back to the vendor for repair.

Crate problems are rare and include power supply and fan failure. Only one backplane problem has occurred.

Network problems are the most worrisome and difficult to isolate. Some crate modules are primarily concerned with communications and a module failure or miswired jumper can cause network problems. These modules include the serial crate controller, the fiber optic U-port, the list sequencer module, and the LAM (pseudo interrupt) encoding module. The configuration switch can be a source of network problems; however, after resolving a number of early troubles, the configuration switch now operates reliably. The serial highway driver is also a possible source of network troubles. The serial highway driver often indicates problems that do not originate in the driver. Central or site software problems are sometimes labeled as network problems.

The equipment used to diagnose problems on the network include the pseudo CYBER channel, the switch probe, the crate analyzer module, and the highway analyzer. The pseudo CYBER channel is a unit that simulates a CYBER channel and allows managing a serial highway driver (a CAMAC network) with an Intel single board computer. The pseudo channel block transfer rate matches the CYBER channel rate and the pseudo channel uses standard CYBER channel cables. The pseudo channel is depicted in its bench configuration in figure 7. It is used on the bench for analysis and development and is being integrated into the configuration switch as an on-line diagnostic tool. The Intel single board computer is being upgraded to a PC-AT.

The highway analyzer has two channels which can be easily patched into the highway ribbon cable in a non-intrusive fashion with a T-connector (see figure 8). When triggered, the analyzer captures a 2K byte time history of highway data. The analyzer uses a single chip microcomputer to format and analyze both protocol and data. It is a very useful instrument and the displays are quite sophisticated. It has, however, limited internal logic to trigger the device. The analyzer will trigger on a parity error on either channel. It also has two external triggers which have been connected to error signals generated by the serial highway driver. In this latter case the instrument must be used near to the serial highway driver.

The crate analyzer is a CAMAC module with 4K bytes of memory which, when triggered, captures the last 400 crate cycles, 80 bits per cycle. Both the crate and highway analyzers use the Intel 8751 microcontroller for control and sophisticated display of the captured data. The triggers now available on the crate analyzer are X and Q bus lines. The pseudo CYBER channel, the highway analyzer, and the crate analyzer were all designed and built in-house at Langley. They are used along with other instruments like standard data analyzers and oscilloscopes to investigate more difficult problems.

A data probe is built into the configuration switch. Any of the 12 highways can be monitored at one of three points on a site card: As the highway leaves the site card for a site, as the highway returns from a site, or as a highway connects to the next site card. Logic in the switch interface unit detects parity problems and loss of clock at the selected probe point. The probe point could be connected to a standard data analyzer. Since the development of the highway analyzer there appears to be no urgent requirement to develop more support for the switch probe.

There are many other items used for diagnostic analysis. Often the fiber optic cable is checked visually using the clock signal, the 1-Km U-port (850 nm), or a flashlight. Occasionally a fiber optic reflectometer is used to check fibers. The indicator lights on the front panel of all the CAMAC modules contain much information. In addition, two off-the-shelf diagnostic modules are found at every Langley site. They are the Data Display (KSC 3291) and the Dataway Display Control (KSC 3296). The 3291 displays the state of the crate bus on indicator lights for each CAMAC cycle. The 3296 is a logic module with a number of toggle switches on the front panel. The 3296 is used to capture a specific Dataway cycle on the 3291 when a particular slot and function are addressed.

For quality assurance every working day at each site the technician tests the site off-line, using programs running on the LSI microcomputers. Some of the tests are simple, effective, and straightforward such as checking meter readings or checking the position of a camera carriage or a motion platform. In other cases, a simplified aircraft model simulation is actually flown in real-time at the site on the microcomputer. Other technicians run tests before the prime shift using the entire ARTS system, including CYBER mainframes, configuration switch, software, sites, etc. Finally, as real-time jobs enter the system requesting service, the static scheduler software configures the simulation network with the configuration switch and then tests each site on the network before releasing the network to the user job.

These various tools and techniques for quality assurance and diagnostic analysis, along with the experience gained with the operational system since mid-1987, have given a high level of confidence in the reliability, maintainability and extendability of the ARTS system.

MEASURED SYSTEM RELIABILITY

Table 5 contains a summary of the trouble reports on the ARTS system for a recent 11-month period. This data was extracted from the trouble reports of the support contractor who maintains the ARTS system. A total of 78 problems occurred during that period or less than 2 per week. This failure rate is expected to decrease in the near term for several reasons discussed here. As shown in the table, 44 percent of the total errors occurred in the highway system, 29 percent in the site equipment, 5 percent in the clock system, and 22 percent in the consoles. This discussion will be limited to the first two systems, except to note that the four Jupiter-12 graphic systems are experiencing a total of one failure per month. Two types of crate modules are responsible for about 33 percent of the problems. These are the one-kilometer fiber optic U-ports and the digital-to-synchro converters. Two improvements have been made in the U-ports which will improve their reliability significantly. First, Kinetic Systems has replaced a discrete component preamplifier on the receiver side with an integrated circuit in all units. The second is procedural; maintenance engineers have learned how to detect problems caused by maladjustment of the phase locked loop in the frequency multiplier on the transmitter side and have learned how to properly adjust the circuit.

There have been several problems with the digital-to-synchro converters which are mounted three Natel Engineering Company units to a CAMAC module. The first problem was a current surge at power-up, which tripped the crowbar circuit in the crate power supply. Kinetic Systems has provided a fix for this problem and these units are now in the process of being rotated through their plant for modification. The modified units are now in service at one site. The second problem with the DSC modules was caused by underestimating the power needed to drive devices for at least one site. The solution planned is to replace the digital-to-synchro modules at particular sites with improved units also built by Natel which have more output power (5 VA) derived directly from the external 400-cycle, 26-volt supply from a motor generator set. These newly developed modules have just been delivered and are currently being evaluated. During the transition period older external analog/synchro units are driven by CAMAC digital-to-analog converters. The third problem is an unexplained failure rate of the DSC modules. There have been 9 failures in 114 channels over the 11-month period. When this problem is solved, the DSC units will meet the high reliability of the rest of the system.

The component failure information in table 5 should be considered along with component quantity data given in tables 1 through 3. The equipment, in general, is powered continuously except for a few weekends a year. Simulations are scheduled five days per week on prime shift and on second shift an average of three days per week. The data shown in table 5 has not been exhaustively analyzed except for a few modules such as the U-ports and DSC modules. Such an analysis would probably not be fruitful. The data was taken during a period when all personnel involved were learning the system. The maximum time to repair was two hours but most problems were resolved in less than an hour. Many of the problems are found during quality assurance tests prior to run time. The number of failures and mean time to repair are both expected to be significantly reduced in the next year.

The system is quite reliable. When troubles occur, they are usually easy to isolate and analyze. The exception to this is trouble with the highway because isolating the fault to a particular network unit in the ring is difficult. With additional tools and additional experience, diagnosing these problems is becoming easier and much faster. Service provided by the primary vendor, Kinetic Systems Corporation, has been competent and responsive.

SIMULATION COMPUTER UPGRADE

Langley is undertaking an improvement in the host computers used for the solving of the equations of motion for flight simulation. This involves the acquisition of new computers and the phasing out of the current two CYBER 175 computers that have served well since 1976. To meet Langley's present and near-term requirements, the new computers will each have four times the CPU power of the present computers and less system latency. A continuing problem with the CYBER computers has been the limited amount of central memory available per simulation application. The new computers will each have 64 megabytes of central memory (less operating system use) available per simulation. The new computers will connect directly to the ARTS I/O system at full highway speed. Each real-time computer will be connected to each of the other real-time computers for communication in real-time at high data rates with low transport delays. Figure 9 shows a transition diagram of the configuration to be used while the new computers are phased into production. The new computers will allow real-time program development simultaneously with real-time flight simulation. The development processors are shown in dashed lines since they may be separate units or may be integral with the real-time processors. In addition to the real-time communication between computers, the communication path labeled LaRCNET is a custom local area network that connects all of the major computers in the Langley computer center as well as numerous mini- and microcomputers throughout Langley. The key to this transition is the ease by which a new computer may be integrated into the ARTS system. Three connections must be made--a CAMAC interface to the configuration switch, and two standard 9600 baud asynchronous RS-232 connections. Recent equipment development indicates that a CAMAC interface will be available. It is anticipated that as many as five separate computers will be acquired during this transition.

CONCLUDING REMARKS

The need for a new real-time flight simulation system for the NASA Langley Research Center was recognized in 1981. During the following 18 months, the requirements for a new system were formalized and a preliminary design was established. The design includes signal conversion, signal distribution, clocking, new control consoles, and supporting software. The design differs radically from the system it replaced in that it uses 10 volts, as opposed to 100 volts, for reference, and the converters are distributed to the simulator sites rather than being centralized. The new signal distribution system is serial digital rather than parallel analog. In addition, each site has either a mini- or microcomputer and in the case of the control console, a three-microcomputer network. The central point of the design is a system of high-performance local area CAMAC networks which can be dynamically configured into almost any arbitrary combination of simulator sites under the control of the flight simulation application job. This design includes significant enhancements to the CAMAC network. The great majority of the system hardware was not commercially available and was designed and built to specification. The CAMAC enhancements developed for this system have had world-wide acceptance.

The first phase of the development and implementation plan was to develop hardware prototypes and demonstrate proof of concept. This phase was successfully completed in December of 1985. Concurrently with the first phase, the final hardware and software was being developed. By the end of 1986, the majority of the production hardware and software was in place. The simulator site conversion effort was then begun and the FORTRAN applications programs were converted to run with the new system and FORTRAN 77. The major system-wide test of running three production piloted aircraft simulation jobs simultaneously was completed in April 1987. Following this test, one simulation mainframe ran the new system while the other mainframe ran the old system to allow time for transition of the application programs and conversion of the simulator sites. The second mainframe was put into operation with the new system in September 1987. Two weeks following this, the communication cables for the old system were chopped and the equipment removed.

Since September 1987 the system has been operational. Most simulator sites were converted by December 1987. The capabilities of this new flight simulation system far exceed those of the system it replaced. Research engineers, simulation application programmers, operators, maintenance personnel, and analysts are extremely pleased with the functionality, ease of use, ease of maintenance, and other aspects of the system. The system is growing with new applications programs and new simulator sites. The ease of integration of sophisticated new sites is impressive.

Overall experience with the system since it has become operational has been very positive. With the aid of the vendor, early problems with the configuration switch electronics, CYBER to CAMAC interface, fiber optic converters, and digital-to-synchro converters were quickly solved. Factory redesigned units were placed into service and have proved to be reliable.

The future of the system is bright. The system is extremely flexible and simulator sites and simulation computers can be added or deleted easily. With the wide bandwidth and extremely low latency, the system is expected to provide a high-quality flight simulation capability to the turn of the century.

Acknowledgment

The work described in this paper was accomplished by a team of Langley in-house engineers and analysts from the Computer Systems Branch and the Analysis and Simulation Branch. The team included NASA personnel and support contractors from the Unisys Corporation, and one analyst from Control Data Corporation. All the network equipment, the configuration switch matrix, and the signal conversion equipment were designed and built by Kinetic Systems Corporation. Their efforts were critical to the success achieved.

References

1. Crawford, D. J. and Cleveland, J. I. II, "The New Langley Research Center Advanced Real-Time Simulation (ARTS) System," AIAA Paper 86-2680, October 1986.

2. Crawford, D. J. and Cleveland, J. I. II, "The Langley Advanced Real-Time Simulation (ARTS) System," AIAA Journal of Aircraft, Vol 25, No. 2, February 1988, pp. 170-177.

3. ANSI/IEEE Standards 583, 595, and 675, Institute of Electrical and Electronic Engineers, 1976.

4. Cleary, R. T., "Enhanced CAMAC Serial Highway System," presented at the IEEE Nuclear Science Symposium, San Francisco, California, October 23-25, 1985.

	In-Service	Planned
Switch Site Positions	20	36
Total Sites	16	22
Control Consoles	4	4
Simulator Cockpits	9	9
Display Generators	1	5
Laboratories (Air Traffic Control, etc.)	4	6
Sites with Minicomputers	4	11-13
Sites with Microcomputers	12	12
Serial Highway Drivers	11	12
1 Km U-ports	50	64
3 Km U-ports	6	6
20 Km U-ports	-	6

Table 1
Quantities of ARTS Network Equipment

	Total Number of Modules	Typical Number of Modules/Site
LSI 11/73 Microcomputer	16	1
List Sequencer Module	48	3
Serial Crate Controller	50	3
LAM Encoder	15	1
Auxiliary Crate Controller	31	3
GPIB	3	-
RS232	14	1

Table 2
Site Equipment

Units	Channels per module	Total Number of channels	Typical channels per site	Approximate Cost per Channel
ADC	6	390	30	$670
DI	48	1392	96	$20
DAC	6	798	60	$360
DSC	3 *	117	15	$1300
DO	24	1128	72	$40

*Double Width Module

Table 3
Conversion Equipment Usage

Site	Computer	Bus	Status
ACS	VAX 8650	Unibus	In Service
MOTAS (2)	PDP 11/34 PDP 11/44	Unibus	In Service
Cockpit graphics (3)			Planned
GCI (2)	Concept	Gould	In Development
EASILY	M 68010	VME/ARINC	In Service
Building 1293	VAX 11/780	Unibus/Mil Std 1553	Planned
Building 1298	VAX 11/780	Unibus	In Service
Building 1232	M68000	VME	Planned
Robotics Lab	VAX 11/750	Unibus	Planned
Laser Target Generators (3)		Multibus	Planned

Table 4
ARTS Site Minicomputers

Sites	No. of Malfunctions	Highway	No. of Malfunctions
Discrete Output	9	U-port	17
Discrete Input	3	Switch	2
Analog to Digital	1	Serial Highway Driver	3
Digital to Analog	2	Serial Crate Controller	2
Digital to Synchro	9	Auxilliary Crate Controller	3
RS-232	-	List Sequencer Module	5
UMIM	1	LAM Encoder	2
QMIM	-		
RAM	2	Total	34
FIFO	-		
Crate Power Supplies	1		
Total	23	**Console**	
		Disk System	4
Clock		J12 Graphics	12
		LSI 11/73 Computer	1
Site Interface (SCIU)	4	Turret	-
Central	-		
		Total	17
Total	4		
	Total Malfunctions 78		

Table 5
ARTS Malfunction Summary Report
July 31, 1987-July 1, 1988

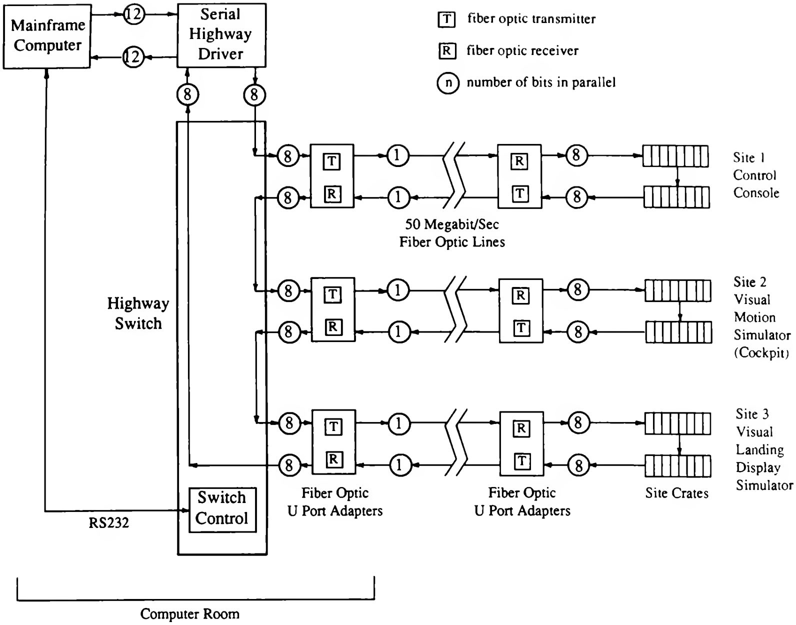

Mainframe Computer
12
12
Serial Highway Driver
8
8
T fiber optic transmitter
R fiber optic receiver
n number of bits in parallel
Highway Switch
Switch Control
RS232
T
R
1
50 Megabit/Sec Fiber Optic Lines
Site 1 Control Console
Site 2 Visual Motion Simulator (Cockpit)
Site 3 Visual Landing Display Simulator
Fiber Optic U Port Adapters
Fiber Optic U Port Adapters
Site Crates
Computer Room

Figure 1
Typical Simulation Network

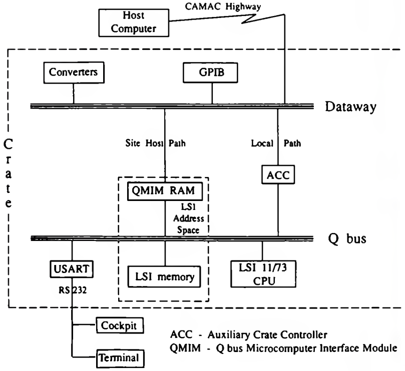


Figure 2
Schematic of LSI at Crate

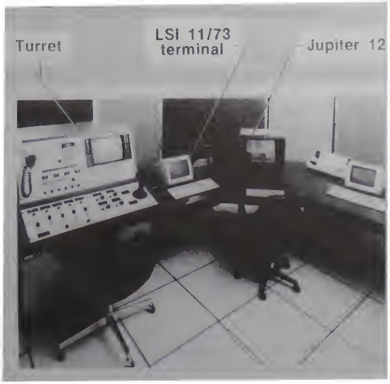


Figure 3
Console Components

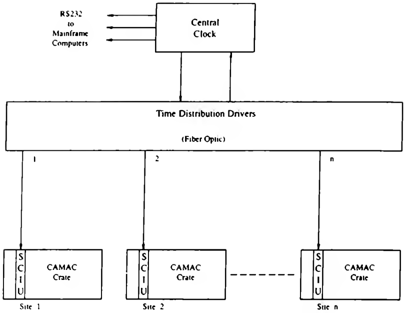
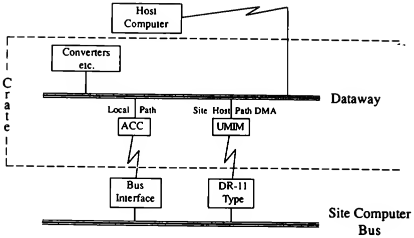
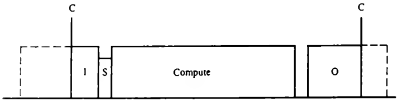


Figure 4
Clock System Block Diagram



ACC - Auxiliary Crate Controller
UMIM - Minicomputer Interface Module

Figure 6
Integration of the Site Minicomputer into Network



C occurrence of interval timer on CAMAC network
I transfer input from CAMAC network to primary memory
S interrupt CPU and start simulation application
Compute compute the equations of motion
O transfer output from primary memory to CAMAC network

Figure 5
Flight Simulation Basic Timing Diagram

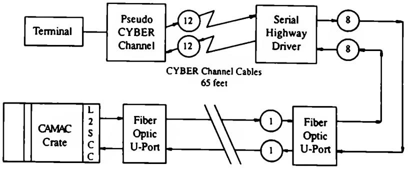


Figure 7
Test Highway on Bench

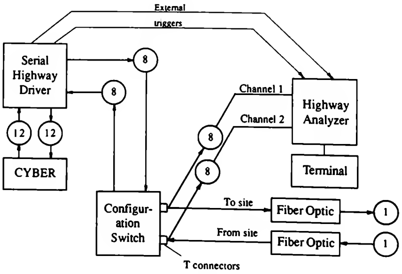


Figure 8
Serial Highway Analyzer

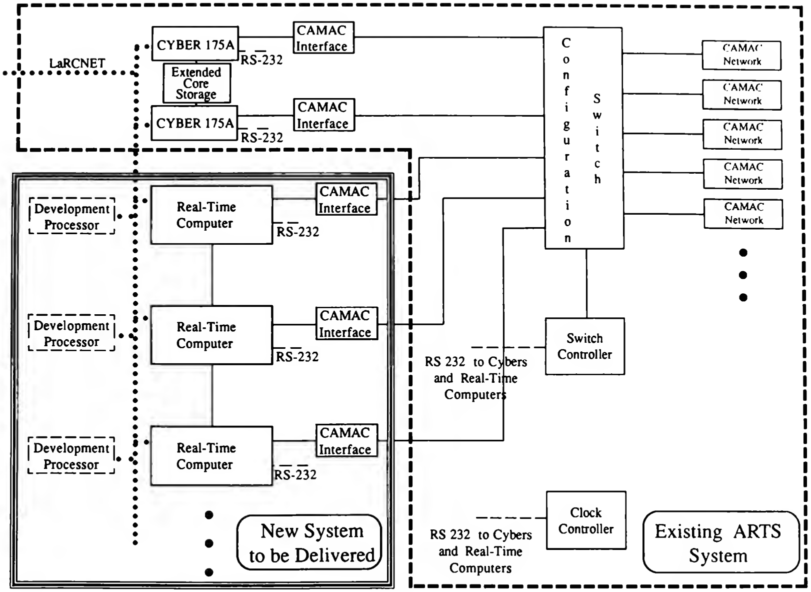


Figure 9
Transition Configuration

Biography

Daniel J. Crawford is the Project Manager of the Langley ARTS System Project. He received a B.S. in Physics in 1961 from the University of Massachusetts at Amherst and a M.S. in Electrical Engineering in 1975 from The George Washington University. He has worked in the field of flight simulation at Langley since 1962.

Jeff I. Cleveland II is the Software Project Manager of the Langley ARTS System Project. He received a B.S. in Electrical Engineering in 1963 from Texas A&I University and a M.S. in Electrical Engineering in 1970 from The George Washington University. He has worked in the field of flight simulation and operating system software since 1963. He also is Adjunct Professor of Computer Science at Hampton University.

Richard O. Staib is a Senior Research Engineer at Langley. He received a B.S. in Electrical Engineering in 1956 from Purdue University. He has worked in the field of Data Acquisition Systems design and development since 1956. He is responsible for all of the locally developed highway and diagnostic equipment for the ARTS System Project.

A STUDY OF THE EFFECTS OF DELAY TIMES IN A DOME-TO-DOME SIMULATION LINK

88-4598-CP

L. Johns*
Northrop Corporation, Aircraft Division
Hawthorne, California

Abstract

There is increasing interest in demonstrating the feasibility of networking flight simulation centers for the purpose of conducting large-scale warfighting exercises. Adverse communication delay effects must be understood and counteracted when possible. In this study, two onsite domed simulation centers are linked and a range of delay times are imposed. Both gun and missile trials are conducted one versus one with simultaneous weapon scoring done for both engagement geometries. Tests of a delay compensation technique are included. Statistical paired comparisons are made and effects on the weapon models and pilots are discussed. Finally, conclusions favorable to the continuation of networking efforts are made.

Introduction

This document is the report of a study of the effects of communication link delay times, such as would be encountered when two or more simulation centers are connected during the course of a real-time air combat engagement. The study focused on air combat maneuvering, based on the expectation that such maneuvering would most clearly indicate any adverse delay effects. This expectation derived from the importance of the constant interplay of pilot action and reaction, and from the potential for highly accelerated maneuvers that result in large aircraft state vector changes within relatively short periods of time.

This study was conducted in the Northrop Corporation, Aircraft Division, Integrated Simulation Systems Laboratory (ISSL) between two domed systems, using artificially imposed delay times.

Objectives

In a dome-to-dome link with weapon modeling done within the attacker's simulation system, as would normally be the case, any delay induced difference between the two system states in real time would tend to work to the advantage of the attacking aircraft. This is because the defending aircraft would suffer a two part disadvantage: first, the defending pilot and his avionics sensor systems would suffer delay in what is sensed of the attack, and second, the defending aircraft's evasion and countermeasure actions would be delayed in arriving in the attacker's system. Because weapon scoring would be done according only to the state within the attacker's system, scoring fairness could be affected, depending on the amount of difference.

Alternatively, a system designer might choose to have weapon modeling done within the defender's system, in which case the same fairness issues would apply in reverse.

These tests had the goals of observing the threshold at which adverse delay effects occur and of observing how the adverse effects developed as that threshold was exceeded. The overall objective of this study was to quantify and analyze the effects of these delay induced system-to-system differences and to make some conclusions about the fairness of simulated engagements conducted under these circumstances.

Impact of Delay Times on Engagement Geometry

The first point of interest was in determining the frame-by-frame difference between the two systems, for each test condition under study, in the stored physical state of the aircraft. This difference is given as the sample mean of the differences between the systems in attacker-to-defender geometry, in terms of off-bore angle (OBA) and range. This data indicates, with good statistical power, the basic dependence of system-to-system differences on time delay.

Figures 1 and 2 illustrate examples of the differences in geometry that could exist, due to communication delay, in their respective simulation systems at some moment in time. Each participant, attacker and defender, is faced with an opponent whose presented position and orientation is not current but instead represents the opponent's state back along his flight path at a previous time. In the highly dynamic environment of air combat maneuvering, the difference between the two geometries will vary in total and will also vary in distribution between off-bore angle difference and range difference. Figure 1 shows a situation where both off-bore angle and range differences exist between the two perceived geometries. Figure 2(a) shows a tail chase situation where the difference is in range only. Figure 2(b) shows a head-on situation where there is no perceived difference in either off-bore angle or range although the two geometries are offset in space.

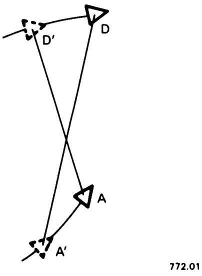
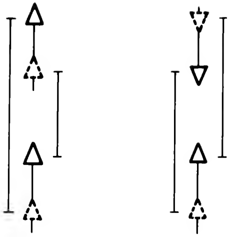


FIGURE 1. DIFFERING GEOMETRIES

*Senior Engineer, Tactical Simulation



(a) TAIL CHASE (b) HEAD ON 772.02

FIGURE 2. TAIL CHASE, HEAD ON GEOMETRIES

Impact of Delay Times on Weapon Scoring

Because high order measures of effectiveness (e.g., loss rates, exchange rates) are an important means of analyzing full-mission simulations, it was decided to attempt to directly gauge the effects of delay times on weapon scoring. To this end, the mean differences between systems in missile and gun scores for each test condition are reported. To allow the comparison to be made, weapon flyout and endgame events were simultaneously simulated within each system in real time, according to the separate engagement geometry that developed within each system. Referring again to Figure 1, when attacker A released his weapon against the perceived defender D', a simultaneous release was made by attacker A' on defender D.

Utility of Data Age Adjustment

Anticipating that adverse delay effects would be evident, the utility of one suggested compensation technique was tested. This technique adjusted the aircraft state vectors received from each sending site to the mission time current in each receiving site. This adjustment for data age was based on a constant-acceleration extrapolation of velocity and position and a constant-rotation extrapolation of attitude. This technique was expected to reduce the difference between the system geometries, to the extent that the underlying assumptions of constant acceleration and rotation were true. The test results allow an evaluation of this expected ability to conform the two system states in real time and to restore scoring fairness.

Design

The briefest test segments appropriate to the study's objectives were chosen so that a statistically meaningful number of trials could more readily be accomplished. Basic fighter maneuvers, with assigned attacker and constraints applied to the defender's options, formed the basis for the tests. These segments – tracking gun shot and AIM-9 missile shots – met the briefness goal and also the goals of emphasis on action/reaction and accelerated maneuvering.

Test Matrix Description

MISSILE TRIALS								
DELAY TIME (sec)	0.25		0.5		0.75		1.0	
DATA AGE	U	C	U	C	U	C	U	C

GUN TRIALS								
DELAY TIME (sec)	0.1		0.25		0.3		0.35	
DATA AGE	U	C	U	C	U	C	U	C

U (UNCORRECTED) = WITHOUT DATA AGE ADJUSTMENT
C (CORRECTED) = WITH DATA AGE ADJUSTMENT
772.03

FIGURE 3. TEST MATRIX

Sixteen test conditions (Figure 3) were tested, encompassing all possible combinations of four delay times per weapon type, two weapon types, and two data age environments. Twelve trials were conducted for each test condition.

The selection of delay times to be tested was intended to span an estimate of the delays imposed by real-world data link options. Any proposed linked system will suffer the delays of local data buffering and latency, and the ordinary transport delays associated with real-time simulation. Encryption/decryption delay may also be necessary. In addition, the system may include a long-haul satellite link involving 270 milliseconds of end-to-end propagation time or the system may be part of a long-haul packet switching network involving as much as 500 milliseconds of propagation time. The selected delay times for these trials were imposed by a software mechanism that added delay to the inherent dome-to-dome link delay. The dome-to-dome link delay was itself controlled by simulation load module configuration to be, in effect, zero. The ability of the test environment to produce identical geometries in the zero delay case was verified prior to formal testing.

The first several missile trials were run uncorrected at each of the four preselected time delays of 0.25, 0.5, 0.75, and 1.0 seconds. This allowed an early look at the effects of the particular delay times chosen for study. For the missile trials, that range of delays fulfilled the test objective and required no adjustment.

When the gun trials were begun, the greater sensitivity of the gun model to attacker-defender geometry required some bracketing to find the range containing the adverse effects threshold as specified for these tests. The detection criterion chosen to define an adverse scoring effect was a system-to-system difference in computed probability of kill of 0.10. The first gun trials were done with 0.25 second of uncorrected delay. The mean of the scoring differences was far greater than the detection criterion. In an attempt to locate this threshold, a set of trials was done with 0.1 second of delay. The mean of the scoring difference was again found to be far above 0.10, so no further uncorrected gun trials were done. With the correction technique enabled, the threshold was bracketed somewhere between 0.3 and 0.35 seconds of delay. For the gun trials, the delay times were 0.1, 0.25, 0.3, and 0.35 seconds.

Two data age environments were tested. In the uncorrected trials, each simulation center computed with no compensation for the age of the state vectors received from the other system. In the corrected trials, within each system aircraft state, vectors received from the other site were updated from their original time of calculation to the current mission time before distribution to the model software.

Two weapon types were tested: the M61A gun and the AIM-9L missile.

Trial Descriptions

In addition to listing trial initial conditions, this section notes some of the considerations taken in devising the trials and certain limitations that were placed on the free maneuvering of the defender.

The gun trials were conducted at 15,000 feet, an altitude chosen to allow good maneuverability by the attacker. Evasive maneuvering proved extremely difficult for the attacker to track and limited the attacker to slashing bursts only. To make tracking possible in this set of trials and thereby produce the necessary amount of scoring data, the defender was limited to a constant g turn governed between two and four g's.

Initial conditions for the AIM-9 missile shot trials were chosen to uniformly span a range of kill probabilities as determined by examination of the kinematic and IR boundaries ("kill envelope") for the AIM-9 missile at the chosen altitude of 10,000 feet. The boundaries plot specified the defender's maneuver to be a 4g turn at constant altitude. Because the simulation did not provide cueing, visual or otherwise, of the launch or approach of an AIM-9 missile, the defender performed the specified constant g maneuver upon the attacker's verbal call of missile launch, which occurred as soon as practicable following trial start. No timed hard maneuvering was possible at endgame.

Facility Description

The ISSL facility is shown in Figure 4. ISSL Dome System 3 was used as the attacker's system for the study. Dome 3 consists of a 28-foot diameter dome featuring 360 degrees of digitally generated color ground scenery. Two General Electric Compuscene III systems provide all-around viewing through light valves and also digitally generate three high-detail projected targets.

The system is equipped with a generic cockpit with a 20-degree field of view monochromatic stroke head-up display (HUD) and two 6-inch monochromatic stroke head-down displays.

ISSL Dome System 1 was used as the defender's system for the study. Dome 1 is a 24-foot diameter dome with a projected 360-degree earth/sky horizon and a nose fixed 150- by 50-degree high-detail projected ground scene. The high-detail ground scene is provided by a Singer Link digital image system. Dome 1 also has a high-detail, 360-degree projected target with video provided from a target model box.

The Dome 1 cockpit is a high fidelity F-20 cockpit with a monochromatic stroke 20-degree field of view HUD and two 6-inch monochromatic stroke head-down displays.

Model Descriptions

The M61A gun model and the AIM-9L missile model used in the study are high fidelity, U.S. Air Force verified real-time models developed by Perceptronics, Inc. The models include detailed flyout and endgame features.

An advanced generic airframe has been implemented in both simulation systems to approximate a next generation fighter capability. The airframe represents an enhanced F-20.

Participants

Aircraft Division employees with appropriate military flying experience and familiarity with the ISSL environment piloted the study aircraft. No special training or familiarization was required.

Evaluation

An evaluation of the test results is restricted to comparison of statistics developed from the paired sample data. There are no absolute thresholds or standards applicable to these comparisons,

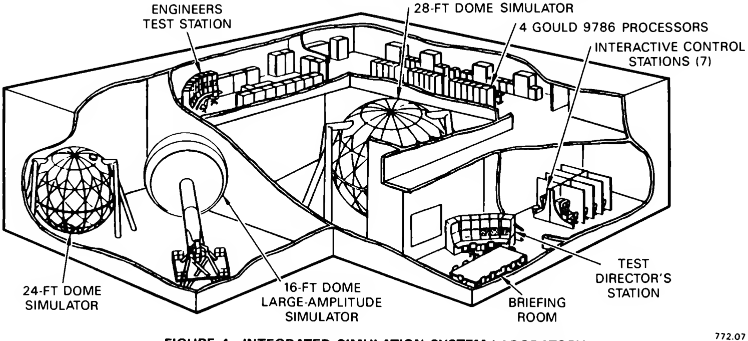


FIGURE 4. INTEGRATED SIMULATION SYSTEM LABORATORY

so an arbitrary detection criterion was used both to allow a determination of sufficient sample size and as a guide in the selection of the range of delay times. The mean difference in probability of kill to be detected was chosen to be 0.10.

The technique of paired comparisons was used, in all of the categories of interest, to statistically quantify the amount of delay induced difference between the two systems. In this technique, X and Y are independent random variables, sampled as pairs, and (X(i) - Y(i)) is the random variable of interest.

System-to-system difference in attacker-to-defender off-bore angle and range were recorded at 1 second intervals. This was the basic parameter used to quantify the amount of difference in attacker-to-defender geometry between the two systems under each set of test conditions. Statistics developed from this large sample serve as the primary means for evaluation of the data age correction technique. These statistics secondarily serve as a check to validate the results of the comparisons of weapon scoring between systems. Increased system-to-system differences in scoring were expected to, and did in fact, track well with increased geometry differences.

The probability of kill for each trial was recorded for both simulation systems. Paired statistics were developed to allow the difference between systems to be directly evaluated. To illustrate by example, Figure 5 contains the actual probability of kill data and geometry difference data for the 12 AIM-9 missile trials involving 0.5 second of delay, no correction technique applied.

TRIAL	ATTACKER'S GEOMETRY	DEFENDER'S GEOMETRY	DIFFERENCE
1	0.984	0.984	0.000
2	0.472	0.462	0.010
3	0.473	0.463	0.010
4	0.434	0.405	0.029
5	0.981	0.984	-0.003
6	0.860	1.000	-0.140
7	0.730	0.870	-0.140
8	0.650	0.450	0.200
9	0.000	0.000	0.000
10	0.760	0.830	-0.070
11	0.750	0.730	0.020
12	1.000	1.000	0.000

MEAN P_k DIFFERENCE = 0.01 STANDARD DEVIATION = 0.09
MEAN RANGE DIFFERENCE = 541.0 ft
MEAN OFF BORE ANGLE DIFFERENCE = 1.49 deg

772.04

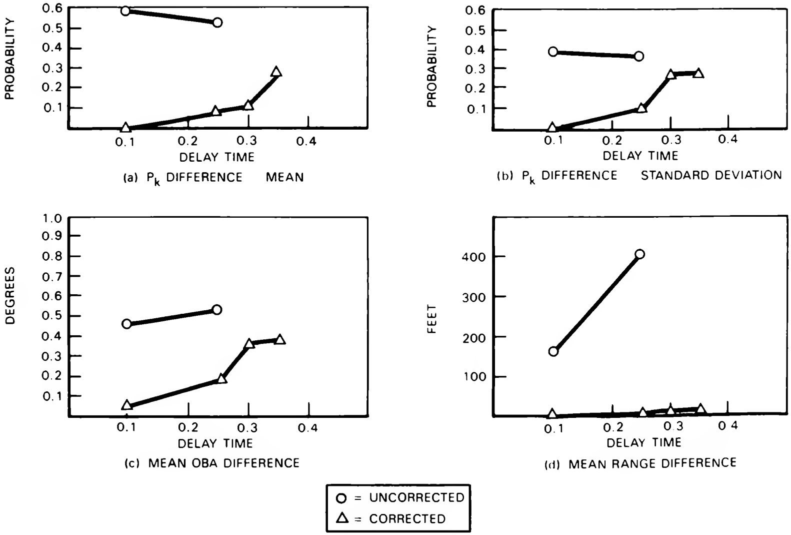
FIGURE 5. AIM-9 RESULTS (0.5 SECOND DELAY, NO CORRECTION)

Results

Figure 6 contains plots of the gun statistics; Figure 7 contains the missile statistics.

The statistics presented are:

1. P(Kill) – for the pair of probability of kill values associated with a test trial, compute the difference. For all of the differences for all of the trials of the test condition, compute the sample mean (Figures 6a, 7a) and the sample standard deviation (Figures 6b, 7b).



(a) P_k DIFFERENCE MEAN

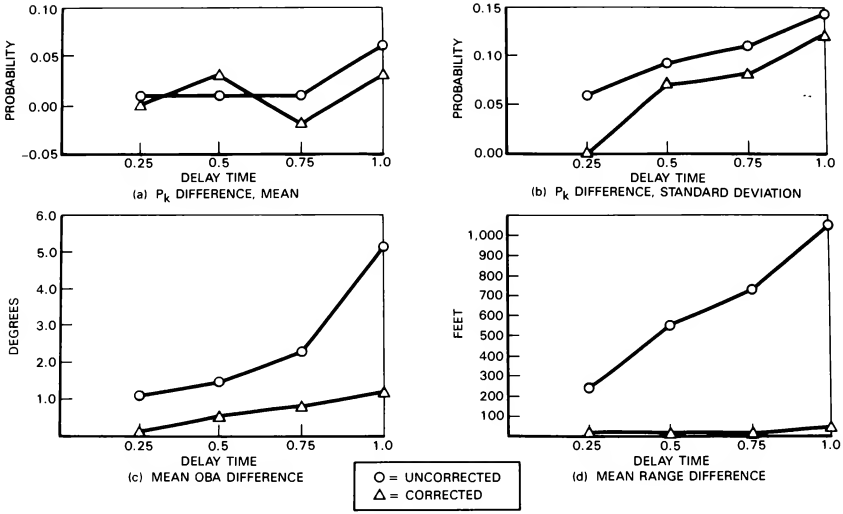
(b) P_k DIFFERENCE STANDARD DEVIATION

(c) MEAN OBA DIFFERENCE

(d) MEAN RANGE DIFFERENCE

772.05

FIGURE 6. GUNNING STATISTICS



(a) P_k DIFFERENCE, MEAN

(b) P_k DIFFERENCE, STANDARD DEVIATION

(c) MEAN OBA DIFFERENCE

(d) MEAN RANGE DIFFERENCE

FIGURE 7. AIM-9 MISSILE SHOT STATISTICS

772.06

2. OBA – for all of the system-to-system differences in attacker-to-defender off-bore angle recorded during the trials of the test condition, compute the sample mean (Figures 6c, 7c).

3. Range – for all of the system-to-system differences in attacker-to-defender range recorded during the trials of the test condition, compute the sample mean (Figures 6d, 7d).

Discussion of Results

A discussion of the study results can be organized into comment on two areas: (1) the effects of delay times on the weapons models, this being the direct subject of the study, and (2) the effects of delay times on the pilot participants, as much as can be reasonably inferred from the results. A discussion of the delay effects as perceived by the pilots naturally involves a separate treatment of the attacker and of the defender.

It is worthwhile to note that the issue of effects as perceived by the pilots is inevitably tied to considerations of simulation system fidelity. The potential adverse effects discussed here must all be evaluated with respect to the fidelity of the visual portrayal of aircraft and missiles, the onboard systems related to weapon employment and defense, and the weapon models.

The statistical results can be generally interpreted as follows. The mean of the scoring differences is the indicator of the direction and magnitude of the systematic unfairness in the scoring. A mean of zero indicates no unfairness, a positive mean indicates accumulated unfairness in favor of the attacker, a negative mean indicates unfairness in the defender's favor. The standard deviation of the difference of the scoring results is an indicator in the worst case sense of the magnitude of the delay induced effects.

Effects on the Gun Model

Referring to Figure 6, note that in the uncorrected gun trials no imposed delay time tested, down to 0.1 second, resulted in mean probability of kill differences less than 0.10, the test criterion chosen to define an adverse scoring effect. At 0.1 second of delay, the mean was found to be 0.59. The gun model can itself be said to be intolerant of uncorrected delay times in the range tested.

In the corrected gun trials, the gun model met the test criterion up to a delay time of approximately 0.3 second.

Effects on the Gun Attacker

Two options are available to the system designer confronting the fairness issue. The weapons modeling can be done in either the attacker's system or in the defender's system. The attacker's perception of the weapon employment problem is unaffected by delay if the scoring is done according to his perceived geometry. He will find himself able to track and score with the guns as well as his skill permits, no matter how great the delay induced geometry difference. On the other hand, if the modeling is done as the defender perceives the attack, some magnitude of geometry difference will eventually create performance difficulty for the attacker; what appear to be good tracking solutions will go unrewarded by good scores.

As noted in the discussion of the effects on the gun model, 0.3 second of corrected delay produced a mean scoring difference of 0.12 with a standard deviation of 0.27. The interpretation is that if the gun modeling were performed according to the defender's geometry, the attacker's perceptions would be adversely affected by these objective difference levels. It seams likely that performance difficulties might begin to be noticeable at or near this time delay level. In support of this estimate, note that the mean and standard deviation for 0.25 second of corrected delay are 0.08 and 0.09, which seem to be levels that would not trouble the attacker, while the levels at 0.35 second of corrected delay are 0.28 and 0.28, which in combination do appear potentially troublesome.

Effects on the Gun Defender

It can be reasoned that if 0.25 second of corrected delay is tolerable when working against the attacker, then the defender could tolerate something more than 0.25 second of delay with scoring done according to the attacker's perceived geometry. This assumes that the defender has a task less demanding of precision than the attacker. But whether this assumption is true or not, note that the off-bore angle and range differences even at 0.35 second of corrected delay are 0.39 degree and 5 feet, unlikely to present a perception problem to the defender. In fact, a close tracking exercise not involving use of the gun model, but including 5g evasive maneuvering by the defender, produced mean off-bore angle and range differences of 1.2 degrees and 21 feet at 0.5 second of corrected delay.

Effects on the Missile Model

The missile model was little affected by delay times. Referring to Figure 7, note that the standard deviation of the mean scoring difference between systems is only 0.08 at 0.5 second and 0.11 at 0.75 second of uncorrected delay. The improvement (decrease) in the variances of system-to-system scoring differences in the AIM-9 trials due to the correction technique was not large, but there was not much room for the correction technique to provide improvement. The low impact of uncorrected delay times can be explained by concluding that the missile model used thereby shows itself to be largely insensitive to launch geometry differences in the range of delays and shot geometries tested. The missile model is able to guide and fly itself to similar endgame situations from differing shot situations, and given that the missile trials restricted the defense against the missiles, and in particular did not include last ditch maneuvering timed to evade the missile, this ability statistically dominates the probability of kill results.

Effects on the Missile Attacker

In the missile trials, increasing delay times result in increasing differences between the two geometries in missile scoring, as is evidenced by the associated increasing variances. However, the missile scores are not consistently skewed in favor of either attacker or defender, as is evidenced by the near zero value of the mean of the probability differences. That is to say, in the range of delay times tested, the attacker was not able to realize a scoring advantage over the defender. The higher missile score was as likely to be achieved in the defender's geometry as in the attacker's geometry; the attacker is as likely to make a better launch in the geometry that the defender sees as in his own presented geometry. Only at 1 second of delay in the no correction case is there any indication of a trend in favor of the attacker, any indication of the attacker exploiting the shot geometry difference. These results can be explained by concluding that the maneuver and launch process carried out by the attacker is inherently imperfect within a tolerance not exceeded by the delay induced geometry differences.

Effects on the Missile Defender

No statistical data were obtained that support direct discussion of the effects on the missile defender. Instead, because the defender had no perception of the incoming missile, the tests were structured to explore the effects on the attacker and the missile. To address concerns related to the defender's missile avoidance problem, the following is offered as the basis for a conservative recommendation.

Certainly the timing of a last ditch effort to defeat an approaching missile is critical, and the standard deviations of the probability of kill results are large enough at the higher delay times to indicate that those worst cases exist where an evasion attempt could be successful in one system and not in the other. Importantly, this is true even in the corrected cases, even though the geometry differences were negligible in those cases as seen in the off-bore angle and range differences. It can be reasoned that if the defender had maneuvered freely and with a perceived basis for timing those maneuvers, even those small delay induced geometry differences might have, at some magnitude within the range tested, become significant to the evasion process. It is reasonable to expect that, with sufficient cueing for missile avoidance, issues of pilot/system reaction and response times (and communication delay times) are likely to be critical to the quality of the defensive flight path. With missile modeling done in the defender's system, which presents no evident problem to the attacker, these concerns would be answered.

Conclusions

Gun engagements with time delays in the range tested require that the correction technique be used and that the weapon modeling be done according to the attacker's perceived geometry. In this case, a conservative estimate is that 0.3 second of delay can be tolerated by the defender.

The missile model, from the scoring standpoint, is little affected by delay induced geometries produced in the range of delays tested. The missile attacker also appears little affected by delay times.

In a system where the defender has use of a high fidelity visual portrayal of an oncoming missile, or a simulated onboard evasion system, it may be prudent to model the missile in the defender's system.

THE APPLICATION OF MANNED SIMULATION TO OPERATIONAL TEST AND EVALUATION: A LOGICAL EXTENSION OF THE TREND TOWARD THE MODULAR DEVELOPMENT AND INTEGRATION OF TRAINING AND ENGINEERING SIMULATION DURING WEAPON SYSTEM DESIGN

88-4600-CP

Ronald G. Hughes, PhD*
McDonnell Douglas Helicopter Company
Mesa, Arizona

Abstract

It is a well recognized fact that conventional range system concepts and facilities are becoming less capable of providing an adequate tactical mission environment for the operational test and evaluation of new weapon systems. At the same time, there is a growing awareness (largely within the training community) of the potential of manned simulation for effectively capturing the fidelity of tomorrow's battlefield environment and for doing so in ways that more effectively integrate the use of advanced simulation and range system concepts. With programs such as LHX, ATF, etc., we are seeing a level of manned simulation inherent to the weapon system design process itself that can be used to significantly extend conventional test and evaluation resources and methods ... if properly conceived and managed. One such concept for doing so is described in this paper. The concept assumes the presence of a structured and highly integrated approach to the use of manned simulation throughout the life cycle of a weapon system. The concept builds upon recognized approaches to the modular design and development of simulators, and it makes a clear distinction between the Industry item-under-test and the Government responsibility for providing a robust and uniform test environment for that item.

Introduction

TEACS is an Army concept for "Test Enhancement Through the Application of Combat Simulation." The concept suggests that a number of the current and projected deficiencies associated with the use of "traditional" operational test and evaluation (OT&E) resources/methods can be overcome through the systematic and routine use of manned simulation during the OT&E process. The present paper presents a conceptual framework for how this might be accomplished.

Under the TEACS concept, manned simulation would be used to augment and not to replace the use of conventional test and evaluation resources (e.g., ranges, maneuver areas, etc.). While the use of manned simulation for test and evaluation is not new (e.g., Perkins and Passmore, 1982), an overall concept for integrating the use of simulation across engineering, training, and OT&E areas has not been put forth. We believe the concept discussed here has the potential for leading to such an integrated approach and, as such, possesses significance not only for the Army Materiel Command (AMC) but for the Department of Defense as a whole.

Background

Current Test/Evaluation Resources are Becoming Increasingly Inadequate

The traditional instrumented range environment under which we subject systems to test is inherently limited in its ability to provide control over many of the key variables affecting weapon system effectiveness under combat conditions. Given the increasingly complex nature of the modern battlefield, these "limitations" are becoming more serious rather than less so (BDM, 1981; SRI, 1981, 1984).

Take, for instance, the need to provide an adequate operational test environment for a system such as that being developed under DARPA's "Pilot's Associate" program. An adequate operational test of the Pilot's Associate concept is more than a simple test of the electronics themselves. The test is basically one of "given the full complexity of the intended mission environment, can the PA system arrive at tactically appropriate 'conclusions' under the constraints of real time combat contingencies?" Where can the real time contingencies of actual combat be represented at a level of complexity sufficient to provide a robust test of the PA system concept? We would assert that manned simulation represents the only way to go beyond the constraints of the traditional operational test environment to a level where such advanced system concepts can be properly evaluated.

There are also examples in the realm of conventional air- and ground-based systems where one has been forced to use simulation in lieu of field test conditions to derive estimates of weapon system effectiveness. Specifically, we are referring to experimental attempts to estimate the effects upon overall system effectiveness and survivability associated with the presence and use of threat laser systems on the battlefield (Stamper, Randolph, Levine, and Hughes, 1982; Hughes, 1983). Such studies are not possible under conventional conditions for both pragmatic and ethical considerations.

Our own technology has brought us to the point where we are rapidly becoming unable to directly observe key man-machine elements of overall weapon system design except under simulated conditions. Likewise it is becoming impossible for human operators to acquire and to perfect many critical skills except under simulated conditions.

Increasing Use of Manned Simulation

There are a growing number of examples of the use of manned simulation in support of advanced weapon system design and utilization (e.g., the U.S. Army's Advanced Rotorcraft Technology Integration (ARTI) program; the U.S. Air Force

*Senior Staff Scientist, Engineering and Training Simulation Department

Member AIAA

Operational Utility Evaluation (OUE) for the Advanced Medium Range Air-to-Air Missile (AMRAAM); the Air Force Aeromedical Laboratory's development of a "super cockpit," etc.). Programs such as the Army's Light Experimental Helicopter (LHX) program and the Air Force's Advanced Tactical Fighter (ATF) program exemplify the growing importance placed upon the use of manned simulation throughout the design and development of new weapon systems. In short, manned simulation ... at a level of fidelity not formerly associated with the use of simulation for engineering development ... is becoming a commonplace asset or component of the weapon system design process.

The Correlation Between Simulator and Range Data

A number of recent demonstrations have focused on the relationship of manned simulation to operational mission effectiveness (Hughes, Graham, Brooks, Sheen, and Dickens, 1982; Killion, 1986). In the study performed by Hughes, et al, (1982) within the context of training effectiveness, limited (training) trials conducted in the A-10 configuration of the Air Force's Advanced Simulator for Pilot Training (ASPT) increased the survivability of mission qualified A-10 aircrews by 15-20 percentage points on the average as measured under RED FLAG force-on-force conditions. In addition to showing that there was, in fact, an observable relationship between performances in the simulator and those under combat-like conditions on the range, the data also suggested that the simulator might be an extremely cost-effective means for preparing aircrews for participation in test and evaluation exercises conducted under similar force-on-force conditions.

The data reported by Hughes, et al, (1982) also provide the opportunity for several comparisons of interest from the standpoint of the present TEACS focus on manned simulation.

Discrepant Estimates of System Effectiveness and Survivability

First, it was observed that estimates of survivability based upon range performances (i.e., RED FLAG in this case) were significantly higher (by as much as 60 percent) than those based upon simulator performances obtained under comparable (simulated) combat conditions. Hughes, et al, attributed these differences to artificialities in the range environment not present in the simulator. Some aspects of mission fidelity were actually better represented in the simulator than on the range (in particular, the proximity of surface-to-air threats to ground targets for the close air support (CAS) mission). Estimates of survivability obtained through simulation were found, on the average, to correlate better with analytically derived estimates than with estimates derived from operational range conditions.

Both the range and the simulator must be viewed as "simulations" of the actual combat mission environment. While the range permits use of the actual aircraft, the simulator is much less constrained in its simulation of the surrounding tactical environment and its effect upon the aircraft than the conventional range.

Critical Event Frequencies: Simulator Vs. Range

The study also observed that the post-PED FLAG performances of pilots in the simulator were continuing to improve even after repeated exposures to the same "canned" CAS scenario; that is, the performances of mission qualified aircrews had still not reached their highest level following a frequency of mission rehearsal far in excess of that commonly associated with any test and evaluation exercise.

The frequency of exposure to critical scenario events in the simulator relative to the frequency attainable in the "operational" range environment is a critical issue not only for training considerations but for test and evaluation considerations as well. To the extent that the aircrew is a critical component of overall weapon system performance and in many cases a discriminating factor, these data would suggest that our current inability to maximize (or even stabilize) the performance of the human subsystem prior to test results in a weak, inconclusive test of overall system capabilities.

It is a well recognized fact that performances during the initial trials of learning any new task are extremely variable. One is often forced to a conclusion of "no difference" when decisions must be based upon such performances. It is quite likely that many expensive tests of hardware and overall system differences reach a conclusion of "no significant improvement in operational effectiveness" because of this reason alone.

Cost Per Test Event: A Significant Measure of Merit

Elsewhere, Hughes, Polis, Fay, Hines, and Altman (1985) looked at the frequency of threat (surface-to-air missile and/or AAA) engagements per sortie in the simulator for the close air support mission relative to that observed during the nominal RED FLAG sortie. While the event frequency per sortie was not that different (1.5-2.0 on the range versus 3-5 per sortie in the simulator), the simulator produced a significantly higher overall rate due to its ability to generate 12-15 sorties per hour versus a RED FLAG sortie rate of 1-2 sorties per day. Given comparable fidelity across both the range and simulator environments, it is clear that the simulator can provide a much better sampling environment upon which to base conclusions having some acceptable degree of statistical confidence. At a minimum, the simulator represents a clear choice as to cost and training effectiveness for test preparation (training) activities.

Where it is possible to reduce test objectives to specific test "events," cost per test event represents a meaningful measure of merit as to the efficient use of available resources. Hughes, et al, (1985) proceeded to explore the range of such a measure for hypothetical simulator operating costs between $800 and $2600 per hour and range costs per hour estimated on the basis of recognized F-16 flight hour costs (recognizing that aircraft flying hour costs are not the only source of range costs). The simulator was found to be less expensive by a factor of almost two in terms of operating cost per hour. Cost per electronic combat (EC) event (given 2.7 events per sortie hour in the aircraft and 36 events per hour in the simulator [12 sorties times 3 engagements per sortie]) was more costly for the aircraft by a factor of over 20:1.

An Argument for "Equivalent" Range Time

The ability of simulation to compress test events as well as, in some cases, its ability to decrease the cost per test event are both positive points in favor of using simulation to augment the traditional (range-oriented) OT&E process wherever appropriate. It is a well known fact that current range facilities are unable to fully satisfy present test and evaluation requirements. There is every reason to believe that the problem will become worse in the future.

The issue becomes one of how to obtain more "equivalent" range time without significantly increasing the real estate associated with ranges. The Hughes, et al, (1984) argument suggests that under certain circumstances one could achieve a twenty-fold increase in sampling rate (test events per hour) at half the cost of conventional, range-oriented approaches. We believe that such a metric should be applied to actual Army OT&E problems and test events to determine if similar savings could be obtained.

Ranges and Simulators are not Mutually Exclusive

As Hughes, et al, (1985) point out, the range and the manned simulator do not represent mutually exclusive resources for the training and/or assessment of pilot and weapon system performance. In fact, the major point made by the authors is that there are potentially significant benefits to be had by integrating major range and simulator assets (for training purposes).

The integration of advanced simulator and range system assets for the purpose of operational test and evaluation also makes sense. Such integration need not mean physical collocation. We are speaking more of integrating the function of the two much in the same way that it is necessary to functionally integrate range and analytic approaches in terms of their treatment of major test issues and variables.

Consider for a moment that one of the major functions of an engineering simulator during the weapon system design process is to serve as a "surrogate" flight test (i.e., range) environment. The notion expressed by TEACS is simply to extend (for systems under development) the use of the engineering simulator to the operational test phase of weapon system development. Just as the prototype air vehicle goes to the range for test, the engineering simulator might also "go to the range," the simulated range that is, for certain aspects of the test. The notion of an engineering simulator being used to augment operational test is a concept that will be explained more in the sections to follow.

The Need for an Integrated Approach to the Use of Manned Simulation

To say that simulation should play a greater role in the operational test and evaluation of modern weapon systems is to have only partially addressed the OT&E question. How simulation should be integrated with other more traditional, and still appropriate, approaches must be addressed. How the use of high fidelity simulation for OT&E should be integrated with the increasing use of engineering simulation during full scale engineering development (FSED) must also be addressed. How the Government should position itself now with respect to the increasing use of simulation in all areas, not just for OT&E but also for engineering and training simulation, must be defined and well understood. Simulation throughout all phases of weapon system design from concept development, to engineering development, to test and evaluation, to training system support must be carefully integrated.

Long-Range Simulation Utilization Concept

Internally at McDonnell Douglas Helicopter we have given a great deal of thought to the application of manned simulation across the life cycle of the weapon system development process. The current status of our thinking, developed largely independently of TEACS, is pictured in Fig. 1.

The concept shown in the figure depicts a highly modular system with the engineering simulator at the core of the process. The crewstation and related avionics and flight control components are represented in the figure as a dome and a modular bus-type architecture. These components are shown as a single crewstation module networked with three other higher-order modules: system control and measurement, visual/sensor generation and display, and the combat mission environment simulation. These three higher-order modules provide the "environment" in which the engineering simulator matures from pure "simulation" (characterized by use of simulation software and commercially available hardware) to a configuration resembling that of the traditional "hot bench" (actual flight hardware/software). Development of the engineering simulator parallels the development of the traditional hot bench capability. At each point in this parallel development process, a capability exists to tie the engineering simulator to the hot bench such that it is now possible to address key man/machine issues throughout the development process rather than just at the end. The products of full scale engineering development are thus not just the prototype air vehicle but a high fidelity hot bench/manned simulation of the vehicle as well.

Under the MDHC concept, the engineering simulator is developed in such a way as to be suitable for use in preparing (training) the aircrews participating in the operational test. As such, the engineering simulator serves as the "prototype" of the full mission training simulator to follow. The engineering simulator is also developed in such a way to permit it to interface (given appropriate interface definition) to a TEACS-like (simulated) OT&E environment. The

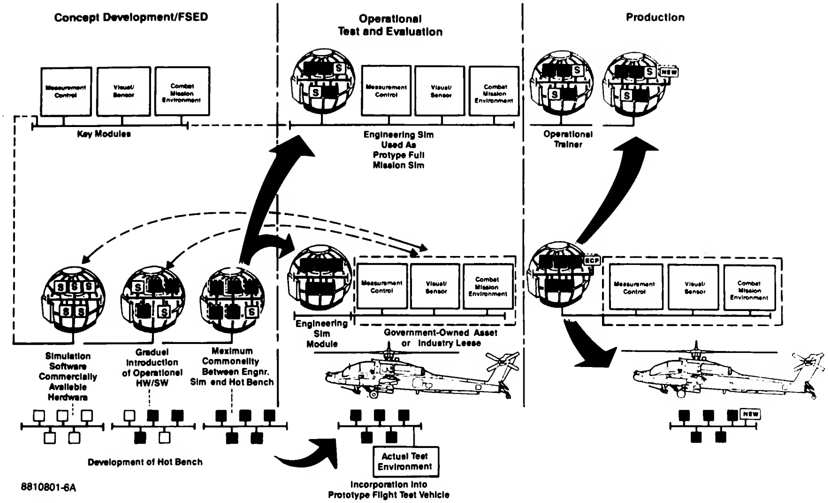

Concept Development/FSED
Operational Test and Evaluation
Production
Measurement Control
Visual/ Sensor
Combat Mission Environment
Key Modules
Engineering Sim Used As Protype Full Mission Sim
Operational Trainer
NEW
ECP
Simulation Software Commercially Available Hardware
Gradual Introduction of Operational HW/SW
Maximum Commonality Between Engnr. Sim and Hot Bench
Engineering Sim Module
Government-Owned Asset or Industry Lease
Development of Hot Bench
Actual Test Environment
Incorporation Into Prototype Flight Test Vehicle
8810801-6A

Fig. 1 MDHC simulator utilization concept.

arrows in the figure from the simulated OT&E capability to earlier phases of FSED indicate the Government's potential for using such a capability throughout the development process to monitor key aspects of the development under mission-like conditions. Following acceptance and the beginning of the production phase, the engineering simulator configuration is retained for future use in the evaluation of Engineering Change Proposals (ECPs). All ECPs would be evaluated first using the engineering simulator (appropriately configured) in the TEACS mission environment maintained by the Government. Once approved, the change is simultaneously incorporated into the aircraft and all training devices.

A Strawman Technical Approach

We begin our definition of a "strawman" system by stating that the overall system architecture of the simulation shall be "modular" in design. By modular, we mean that the overall design of the system shall be largely independent of the design of any of the specific modules; that the link between individual modules and the overall system is contained in the definition of the interface by which modules communicate with other modules and system components over a common bus structure or network. Definition of the interface requirements and definition of the communications network or bus represent important design considerations.

The chief "modules" of the system might consist of the following: the crewstation module to include the simulation of avionics, flight controls, etc.; a system control module containing the means by which the simulator would be controlled; and a module containing the simulation of the mission environment. The chief functions of each of these generic modules are described below.

The Crew Station Module

The crewstation module is viewed as consisting of all elements in the simulation of the immediate cockpit environment of the operator to include the simulation of the missile equipment package (MEP) and flight controls. The number of crewstations shall be dependent upon what is determined to be the smallest tactical element in which the system under test can be most effectively evaluated. It is anticipated that the size of such a tactical element shall nominally be in the range of 3-5 vehicles. It must be remembered that the number of vehicles or elements cannot be taken as being 1:1 with the number of crewstations or "simulators" required. Where the crew configuration consists of multiple crewmembers, the crewstation modules may consist of separate stations for each crewmember to the extent that such an approach does not disrupt essential crew coordination requirements and to the extent that such approaches minimize problems associated with the interface to the visual system module.

The physical fidelity of each crewstation shall depend upon whether that element is the primary element under test or whether that element is represented in the simulation so as to provide a supporting or coordinated function with the primary element under test. Regardless, each crewstation shall exist as a "stand-alone" module capable of being operated under its own power with dedicated onboard computational provisions for flight model computation, avionics processor, controls and displays processor, and all real time input/output

(IO) processing requirements. The crewstation module shall, under ideal conditions, consist of integrated operational hardware and software in an engineering simulation (vehicle hot bench) configuration. The crewstation module shall be transportable and capable of interfacing via a Government-defined interface requirement to the OT&E simulation network. Where operational equipment is resident within the crewstation module, it shall be possible for such equipment to be "stimulated" by input from the simulation environment computer. In the case of sensors, it shall be possible to bypass the front-end of the sensor and to stimulate the system at the level of the display processor or primary pilot interface.

Assumptions

a. High fidelity engineering simulation will be an integral and required part of all future Government-funded full scale engineering development (FSED) programs.

b. The Government will exercise control over the requirements for engineering simulation such that the contractor's engineering simulator shall be capable of being physically transported to the Government OT&E facility and interfaced with that system. The requirement to interface with the Government test and evaluation facility shall exist at all phases of FSED and shall not be limited by program-dependent use of actual equipment or commercially available equipment depending upon the point in system design.

c. The crewstation shall be designed to be motion-worthy for use in six degree of freedom motion environments. Where motion is specified, it shall be the responsibility of the Government to describe any and all interface requirements between the crewstation and motion module.

d. The Government shall establish criteria and procedures whereby the engineering simulation "module" subjected to test can be verified with respect to the degree to which it represents the actual prototype vehicle system under test (for rotorcraft, primarily in terms of flight model and handling qualities).

Environment Simulation

The environment simulation module shall provide the simulation of all friendly and threat weapon system characteristics, for all flyout simulations and their link to the visual system module, and for the tactically appropriate operation of threat weapon systems. The environment simulation module shall be that component of the overall OT&E simulation facility whereby the Government imposes control over the definition of the threat and its capabilities. The relationship of the threat system operation in the environment simulation module to the operation of threats employed (or simulated) in the operational field test portion of the OT&E shall be precisely defined.

System Control Station

The System Control Station (SCS) shall be that module having the function of overall control over the simulation to include the collection and retrieval of data regarding the operation of the system under test. The System Control Station's measurement capability shall exist such that measurement and scoring algorithms resident within the System Control Station module can be applied to data collected either in the simulator or in the operational range environment. The communications network linking together the separate modules of the overall system shall be considered a part of the System Control Station module. It shall be possible from the System Control Station to initialize all elements in the simulation as well as to exert control over elements residing in the environment computer. The System Control Station shall also provide a real time capability for monitoring each element in the simulation and for controlling unmanned elements "flown" from the console.

Visual/Sensor Module

The visual/sensor simulation module shall provide for all visual and sensor simulation provided the individual crewstation modules. From a facilities standpoint, the visual system module shall be designed so as to be able to support the worst case requirements for visual and sensor simulation for five nominal crewstations each having upward to four individual crewmembers. The visual system module shall consist of both image generation and display submodules being able to accommodate the varied seating and viewing conditions of both ground and air vehicles.

Technical Feasibility

The technical feasibility for developing a TEACS-like simulation capability within the context of the Army scout-attack mission has been addressed by Shipley, et al (1988). Using the light platoon of the Attack Helicopter Company as the basic tactical "unit" under test, a simulator baseline system definition was established. The baseline system definition considered the varied requirements of three different mission scenarios: area reconnaissance, close attack, and deep strike. The technologies required to support such a system definition were evaluated both in terms of current capability and integration risk.

A key concern when operating in the rotorcraft nap-of-the-earth (NOE) mission environment continues to be the ability of present computer-image generation (CIG) techniques to adequately support requirements for scene content and object density. A number of visual system "workarounds" are recommended for dealing with current limitations. Even though these workarounds would assure a high level of functional fidelity, in many instances they would require some departure from perceptual, or real world, fidelity. An additional concern revolves around the issue of simulation versus stimulation.

The chief issue involved in establishing a TEACS approach to the scout-attack team requirement was not technology, per se, but rather were budget- and schedule-related. Shipley, et al discuss a number of alternative procurement strategies for the development and utilization of a simulated OT&E capability. One of the more attractive alternatives from the Government's standpoint involved building upon some existing simulator capability and then "leasing" the time required for the test on a time-as-needed basis.

The Relationship Between TEACS and the Current NASA/AMES Crewstation Research and Development Facility (CSRDF) and the DARPA "SIMNET/AIRNET" Concept

The Shipley study addresses both CSRDF and the DARPA SIMNET/AIRNET effort in terms of their relevance to the overall TEACS concept and facilities requirement. The authors point out that the primary mission of CSRDF is to support exploratory research and development (6.2) efforts in the area of rotorcraft crewstation design. The CSRDF has not been designed to accept standalone engineering simulator crewstations of other-than-CSRDF design. Neither has CSRDF been designed to "stimulate" actual aircraft hardware in a manner that might be appropriate under a TEACS approach. What CSRDF does provide, however, in its current configuration is a highly developed simulation of the rotorcraft mission environment to include provision for the activity of "auxiliary" players in addition to the "ownship." In terms of the major TEACS "modules" proposed earlier in this paper, CSRDF is a likely candidate for much of the content of the Mission Environment Module as well as the System Control and Measurement Module.

With respect to SIMNET/AIRNET the following points are made. First, the low cost philosophy that has become synonymous with SIMNET is inconsistent with the crewstation fidelity requirements of TEACS. Even in its "development" (or SIMNET/AIRNET-D) mode, the DARPA approach more resembles CSRDF than TEACS. Both have in common an attempt to simulate the team or collective nature of the tactical mission environment. The two concepts also have in common an ability to network multiple players, a capability essential to TEACS.

In general, then, it was the assessment of Shipley, et al (1988) that neither CSRDF nor SIMNET/AIRNET as each is currently conceived can satisfy TEACS requirements. Both programs, however, provide valuable insights into the provision of the surrounding mission environment and for the networking of auxiliary players in a team or collective performance environment. Of particular interest to TEACS is the manner in which each would deal with the simulation of those elements other than the main element under test. In SIMNET/AIRNET, none of the players in the visually-interactive, immediate tactical environment are "auxiliary" in nature. The level of crewstation fidelity is constant across all players in the immediate environment. In CSRDF, all players outside the simulator, per se, are "auxiliary" in nature and controlled from low fidelity "workstations." What constitutes an adequate environment for controlling these other players is an issue that CSRDF and SIMNET/AIRNET can address.

Summary

Such is our early "strawman" concept of how manned simulation might be integrated into the operational test and evaluation process by way of a deliberate attempt on the part of Government to fully utilize (and deliberately direct) the engineering simulation process employed by the primes as a required part of full scale engineering development. We have tried to point out how such a process might follow a modular approach to the design of major system components. The concept basically puts the Government in the position of specifying and controlling the simulated test "environment" and the manner in which (via its engineering simulators) would interface that environment. Essentially it is a concept for significantly extending the current OT&E test environment through the use of manned simulation. It is a concept for integrating major simulation and range system assets ... not for replacing the present OT&E environment with simulation. The benefits of doing so were described: e.g., significantly reduced cost per test event; increased statistical confidence derived from the ability to collect more observations under a wider range of representative combat conditions; the ability to control and to directly manipulate critical factors not easily controlled or manipulated under conventional testing conditions, etc.

References

BDM Corporation. Threat Training Systems Study. Eglin AFB, FL: Range Instrumentation and Equipment System Program Office. Armament Division, Air Force Systems Command, 1981.

Hughes, R., Graham, D., Brooks, R., Sheen, R., and Dickens, T. Tactical Ground Attack: On the Transfer of Training from Flight Simulator to Operational RED FLAG Exercise. In Proceedings of the Fourth Interservice/Industry Training Equipment Conference and Exhibition, Orlando, FL: 1982. (Also appears in Proceedings of the 26th Annual Human Factors Society Meeting, Seattle, WA: 1982.)

Hughes, R. A Flight Simulator Study to Estimate the Impact of a Directed Energy Threat Upon A-10 Close Air Support (CAS) Mission Performance. AFHRL-TP-85-33. Operations Training Division, Air Force Human Resources Laboratory, Williams AFB, AZ: 1983 (SECRET).

Hughes, R., Polis, D., Fay, R., Hines, J., and Altman, H. The Tactical Training Complex: Prototype Facility for the Integration of Advanced Simulation and Range System Concepts for Tactical Aircrew Training in the 1990s. AFHRL-TR-85-3. Operations Training Division, Air Force Human Resources Laboratory, Williams AFB, AZ: 1985.

Killion, T. Electronic Combat Range Training Effectiveness. AFHRL-TR-86-9. Operations Training Division, Air Force Human Resources Laboratory, Williams AFB, AZ: 1986.

Perkins, J.A., and Passmore, H. Advanced Medium Range Air-to-Air Missile (AMRAAM) Manned Air Combat Simulation (MACS) Operational Utility Evaluation (OUE). Final Report for Contract F-29601-80-C-0044 Prepared for Air Force Test and Evaluation Center, Kirtland AFB, New Mexico: August 1982.

Shipley, B., Blizek, R., Hansen, J., Hughes, R., King, W., and Robertson, G. Scout/Attack Team Test Study: Technical Report. Final Report for Contract DABT60-87-C-4118. Prepared for Directorate of Combat Developments and TRADOC System Manager of Scout Helicopters, U.S. Army Aviation Center, Ft. Rucker, AL: 1988.

SRI International. Threat Training Systems Study, Armament Development and Test Center, Eglin AFB, FL: 1981.SRI International. Electronic Combat Training Range System Study. Briefing for: USAF Threat Simulator Working Group. Menlo Park, CA: 1984.

Stamper, D., Randolph, D., Levine, R., and Hughes, R. Performance Decrements Following Laser Exposure. Letterman Army Institute of Research Technical Note 82-37TN. Ocular Hazards Division, Letterman Army Institute of Research, Presidio of San Francisco, CA: 1982 (SECRET).

TACTICAL AIR COMBAT IN A REAL-TIME MULTIPLE-ENGAGEMENT SIMULATION

88-4601-CP

Jamison Luhn*

Northrop Corporation
Aircraft Division
Hawthorne, California

ABSTRACT

This paper will present the technical approach that the Northrop Corporation Aircraft Division's Integrated Systems Simulation Laboratory has taken to provide the projects at Northrop with an advanced manned interactive multiple-engagement tactical air combat mission simulation. The hardware and software configuration to enable real-time evaluation will be described. In addition, this paper will present representative formats depicting the nature of the data obtained in this facility. There will be a discussion of the expansion that is now in progress to enhance the laboratory's capabilities and increase the computational capacity.

INTRODUCTION

In the early 1980s Northrop's Integrated Systems Simulation Laboratory (ISSL) conducted a proof-of-concept test by linking its two domed simulators and performing the first one-on-one air combat engagement simulation. The potential for expanding this capability was fueled by the activities of other companies in the field. McDonnell Douglas Corporation had completed their AMRAAM Operational Utility Evaluation (OUE) at about that time. The ability to display information relating to multiple aircraft demonstrated by Cubic Corporation on the ACMI ranges provided the ideas for Northrop to expand and to include many of these features using the latest technology. Over the years, models have been developed from the various R&D projects using the laboratory. A capability has evolved to model the aircraft as an integrated system. This integrated simulation includes the avionics systems, weapons, IR and RF signature patterns, control systems, air vehicle performance, and the environment. Combining these systems in a single simulation permits the fine-tuning of the tactical aircraft and the development of its associated weapon systems. Today, any variety of up to nine of these simulated aircraft can be combined using any mix of two domed simulators and seven manned interactive control stations. This multiple-engagement simulation capability now provides users of the facility an environment for the development and evaluation of such diverse techologies as

- o avionics software in the mission environment;
- o pilot/vehicle interface studies;
- o sensor blending/fusion algorithms and fire/ weapon control algorithms;
- o performance designs and measurements in simulated flying/ tactical situations;
- o electronic countermeasure systems;
- o multiple-engagement/internetted tactics;
- o air-to-air, surface-to-air missile delivery and avoidance; and
- o situational awareness.

It is here that design and technology tradeoffs can be conducted in real-time to evaluate/validate aircraft and system designs. Numerous trials can be run, data gathered, concepts examined and modified, initial conditions changed, and the design evaluated again. All this can be done in the course of a day's trials. Northrop's ISSL facility incorporates a flexible modular design in both hardware and software formats. This modular capability allows for different aircraft designs to be examined with minimal impact on down-time or turn-around time, thus sharing assets and schedules among a multitude of projects using the laboratory. When hardware-in-the-loop is anticipated in a simulation, the software is modularized to represent the hardware and its interfaces. The system can be quickly reconfigured to provide any combination of Red and Blue forces, including various sensor, weapon, and airframe mixes. Menu-driven software, controlled by simulation executives, allows the test engineer to build flight simulation modules based on the mission scenario supplied by the test director.

* Senior Engineer, Member AIAA

FACILITY CONFIGURATION

The ISSL facility shown in Figure 1 is comprised of two fixed-based visual dome simulators (Domes 1 and 3), a moving-based visual simulator (Dome 2), seven manned interactive control stations, a Battle Situation Room with a Test Director's Station, and three test engineer stations. The motion-based simulator is not part of the overall multiple-engagement simulation, but is used for stand-alone flying qualities evaluations. Each domed simulator has its own Test Engineer's Station (TES), but the Dome 3 TES controls the multiple-engagement simulations among all the individual flying elements.

Dome 1

Dome 1 is a 24 foot diameter simulator and, when used stand-alone, is primarily for high-detail air-to-ground flight simulations. A General Electric Compuscene IV color graphics generator driven by a Gould Concept 32/9780 provides the high-detail ground scene and target generation. The total field-of-view is 135 degrees by 45 degrees. Pilot head-slaving of this image in two axes is to be incorporated in the near future. For now, an earth/sky projector provides additional visual cues beyond the 135 degrees displayed by the Compuscene IV. This six channel system dedicates three channels for terrain generation and background targets, one channel for projected targets, and the remaining two channels can be used for sensor displays in the cockpit.

Dome 3

Dome 3 is a 28 foot diameter air combat simulator with two three channel General Electric Compuscene III systems that provide a 360 degree-field-of-view color background scene and three high-detail projected targets. Six additional targets can be inserted within the background scene. This dome visual system provides the pilot with a realistic out-the-window aerial combat environment. The Compuscene III system is driven by two Gould 32/9750s.

Cockpit Assemblies

The two fixed-based simulators house removable and interchangeable cockpit/crew stations. The cockpit crew station assemblies (situated in the center of each dome) are representations of modern fighter aircraft. The basic functionally equivalent hardware consists of head-down displays, a head-up display, a stick for pitch and roll control, rudder pedals, and a throttle lever. Some crew stations may contain additional hardware, depending on the technology being investigated. These cockpit assemblies can be changed-out in one working day when projects require their own cockpit configuration.

Manned Interactive Control Stations (MICS)

The MICS, developed by the ISSL, provide high fidelity avionics and weapon systems capabilities with the flexibility to modify or vary parameters during test. Each MICS functions as a separate flight element in the multiple-engagement simulation as shown in Figure 2. The MICS employ a color raster graphic monitor display which combines all the necessary symbology a pilot requires to fly the simulated mission. The MICS also includes a joystick to provide aircraft flight control, weapon firing and avionics mode select, and a thrust control lever (throttle) for

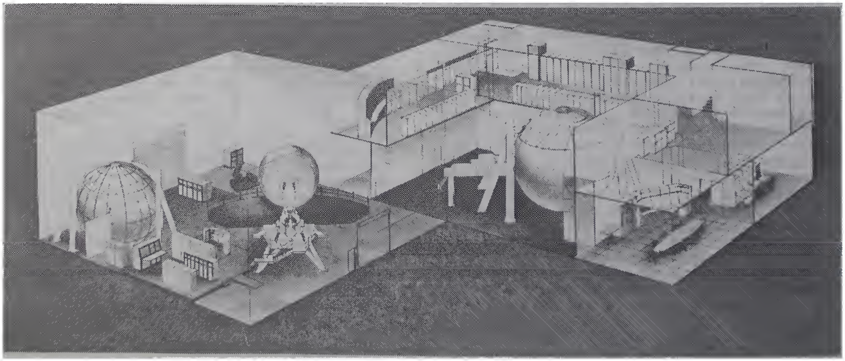


Figure 1. Integrated Systems Simulation Laboratory

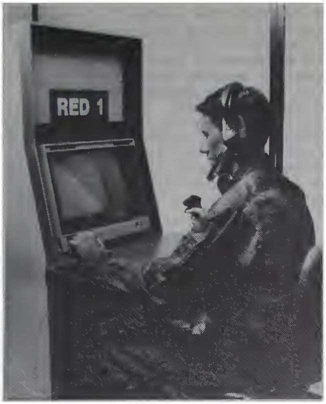


Figure 2. Typical MICS

simulated engine control. A touchscreen controller provides additional avionics and weapon systems control at the station.

Battle Situation Room (BSR) and Test Director's Station (TDS)

The BSR is the primary area for monitoring the progress of the air combat simulations. The TDS (Figure 3) is located in the BSR. From this vantage point, the test director controls the test using a touch sensitive simulation control display that pages through different menus. Features on one page allow the test director to kill off aircraft if they venture into unauthorized flight regions or if they are to be removed from the engagement due to low fuel. Another page of this display permits "dead" players to be regenerated at selected points in the battle area.

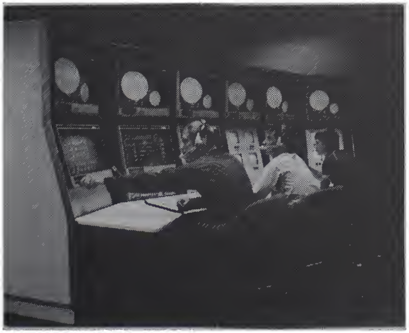


Figure 3. Test Director's Station

The test director also monitors the engagement using a "God's-Eye-View" display, a parametric display, and an array of color repeater monitors for each MICS and the various domed simulator cockpit displays. The "God's-Eye-View" display in Figure 4 presents the test director with the relative position and orientation of all aircraft, the trajectories of in-flight missiles, plus ground-fixed sites such as the FEBA (Forward Edge of the Battle Area), CAP (Combat Air Patrol) points and waypoints. This display can be made to center on any combination of simulation participants as well as be scaled to the desired spatial area. Horizontal, vertical, and perspective views can also be selected during the engagement.

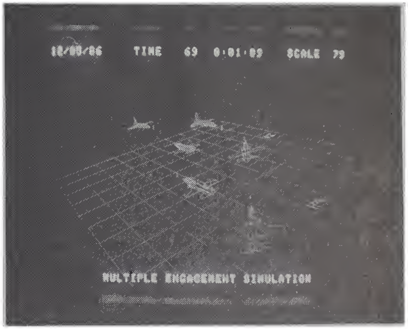

12/08/86 TIME 69 0:01:09 SCALE 79
MULTIPLE ENGAGEMENT SIMULATION

Figure 4. "God's-Eye-View"

The parametric display shows relative information between aircraft such as range, differential altitude, who launched a missile at whom for a kill, radar or IR lockons amongst the aircraft, etc. A computer-controlled intercom network provides secure communication between the team participants. The test director can talk to the Red and Blue teams separately or at the same time for debriefing the test results. Video taping of any display is possible with manual selection by the test director or through computer controlled video switching software.

Other features of the BSR include two large viewing screens, shown on Figure 5. These monitors can be switched between the "God's-Eye-View", para-

Figure 5. Battle Situation Room

metric, and MICS displays. Additional VHS, BETA, or UMATIC video players allow for pilot or guest debriefing even while another group of pilots may be flying the next trial. A computer terminal is also available for reviewing "quick-look" data. This data is a subset of the complete data reduction stored for a particular trial, and it briefly summarizes the weapons employed as a function of target and attacker, weapon type, and outcome. Failure codes are provided in the event a weapon fails to kill its target.

Test Engineer's Station (TDS)

The Dome 3 TES functions as the overall controller and monitor of all the multiple-engagement simulations in the ISSL and the single engagements performed in Dome 3. Some of the functions performed at the TES are the assignment of participants to a particular MICS or simulator; specification of software load modules needed to satisfy the requested simulation test; test initialization, monitor, and control of the simulation and the equipment involved.

Strip chart recorders, magnetic tapes, disk devices, and high-speed printer/plotters monitor the simulation and collect data in real-time to perform post-flight data reduction and analysis. Monitoring of the Red and Blue conversations and pilot comments, as well as secure communication with the test director, is available with the computer-controlled intercom system.

The same color displays presented at the TDS are also available at the TES. In addition, over-the-shoulder cameras are used to display in-cockpit activity on video monitors mounted in the TES control panel. During development periods the control of the "God's-Eye-View" can be switched to the TES. The parametric display presented at the TDS is repeated here for the Test Engineer to monitor during the engagement or to act as a debugging device during development periods. The TES simulation control display has the same capabilities as those at the TDS in addition to initialization displays and simulation control flags. Many of these same capabilities exist at the test engineer stations for Domes 1 and 2.

SOFTWARE CONFIGURATION

The majority of software used in the ISSL multiple-engagement simulation was developed by the laboratory to meet project requirements as well as from non-real-time models that allow man-in-the-loop simulation results to be compared with known standards. A small percentage of software has been acquired from outside vendors. The ISSL places much emphasis on the structuring of its software. Every effort is made to develop the software so that it is easy to modify, and thus accept future expansions. Parameterization of variables (having most routines data driven) enables a vast amount of software developed by the lab to be made available for multiple projects. This is especially true of routines that would normally be classified. Therefore, a project needs only to modify the data to accommodate their requirements. The major software components can be divided into simulation modeling, simulation initialization and monitor, and simulation real-time execution.

Simulation Modeling

The software modeled, summarized in Figure 6, falls into three categories - avionics, airframe, and simulation environment. Avionics models developed in the ISSL simulate modern and futuristic sensors and displays representative of Blue and Red capabilities. These models include electronically scanned array and gimballed multi-mode radars, IR/EO sensors, and a radar warning receiver, all of which can be manipulated with a touchscreen interface by the pilots. The same software models are used for both the MICS and the domes, while separate routines handle the different hardware interfaces. For example, each MICS pilot, functioning as a separate flight element, interfaces with the avionics through a touchscreen attached to a raster color monitor. A joystick controller provides aircraft flight control, weapon firing and avionics mode select while a thrust control lever (throttle) simulates propulsion.

The simple out-the-window display on the CRT screen (Figure 7) provides

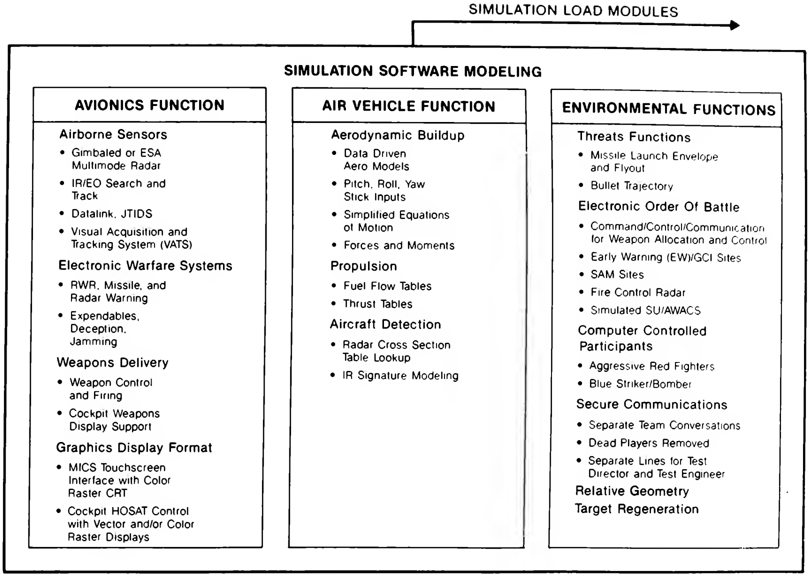

SIMULATION LOAD MODULES
SIMULATION SOFTWARE MODELING
AVIONICS FUNCTION
Airborne Sensors
• Gimbaled or ESA Multimode Radar
• IR/EO Search and Track
• Datalink, JTIDS
• Visual Acquisition and Tracking System (VATS)
Electronic Warfare Systems
• RWR, Missile, and Radar Warning
• Expendables, Deception, Jamming
Weapons Delivery
• Weapon Control and Firing
• Cockpit Weapons Display Support
Graphics Display Format
• MICS Touchscreen Interface with Color Raster CRT
• Cockpit HOSAT Control with Vector and/or Color Raster Displays
AIR VEHICLE FUNCTION
Aerodynamic Buildup
• Data Driven Aero Models
• Pitch, Roll, Yaw Stick Inputs
• Simplified Equations of Motion
• Forces and Moments
Propulsion
• Fuel Flow Tables
• Thrust Tables
Aircraft Detection
• Radar Cross Section Table Lookup
• IR Signature Modeling
ENVIRONMENTAL FUNCTIONS
Threats Functions
• Missile Launch Envelope and Flyout
• Bullet Trajectory
Electronic Order Of Battle
• Command/Control/Communication for Weapon Allocation and Control
• Early Warning (EW)/GCI Sites
• SAM Sites
• Fire Control Radar
• Simulated SU/AWACS
Computer Controlled Participants
• Aggressive Red Fighters
• Blue Striker/Bomber
Secure Communications
• Separate Team Conversations
• Dead Players Removed
• Separate Lines for Test Director and Test Engineer
Relative Geometry
Target Regeneration

Figure 6. Simulation Software Modeling

the MICS pilot with basic flight attitude and altitude information, plus ground cues using a ground grid and an earth-sky pictorial. Visual acquisition software is used at the MICS to simulate a pilot scanning visible portions of the sky with multiple glimpses to detect other aircraft. Using logic based on range, aspect, and visual signature of the aircraft, symbology is provided on the out-the-window display. This symbology, taking into account cockpit masking angles, cues the pilot to look in the direction where targets would normally be seen if not for display field-of-view limits. Visual identification is enhanced with different stick figures to distinguish Red and Blue fighters. This same logic is used to assign aircraft to the target projectors in Domes 1 and 3. Multiple-engagement and internetted tactics are available with data linking and simulated JTIDS modeling. Effects of low observability are evaluated by inserting different radar and IR signature patterns or employing gaining techniques.

Pilots flying the domed simulators are presented these same avionics features on the head-up and head-down displays. The sensors are manipulated with hands-on-stick-and-throttle (HOSAT) controls. Touchscreen displays are also being incorporated into new cockpits. The background scene projected on the dome provides the pilot with spatial cuing, as well as targets within range. With the ability to project targets on the inner dome surfaces, one-on-one visual air combat engagements can be conducted between the two domed simulators.

Air vehicle performance is simulated using a format that permits different aircraft types to be modeled. This feature allows design tradeoffs or mixtures of flight elements to duplicate real-world threats encountered in a multiple-engagement fight. Using simplified equations of motion, the performance characteristics of the various aircraft types can be modeled from data tables. These tables include elements such as maximum roll rate (Q), lift, and drag as a function of mach. Additional thrust and fuel flow tables based on power setting, mach, and altitude combine with the above data to compute the rotational velocities P, Q, and R. Time lags and gravitational and velocity limits add to the fidelity of the model.

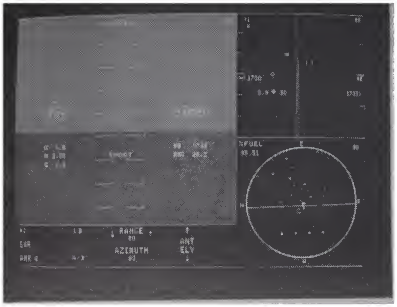


Figure 7. Sample MICS Display

Radar cross sectional area patterns are modeled in terms of azimuth and elevation breakpoints, and multiplicative gain factors. The IR signature pattern consists of six components: airframe, inlet, deck, turbine, nozzle, and plume. Each component has a radiance value (as a function of target altitude and mach) and an area value based on relative geometry (azimuth and elevation) between corresponding aircraft. Atmospheric transmissivity, background clutter, and ownship IR sensor sensitivity are also taken into account.

As part of the simulation environment, air-to-air weapon delivery and fire/weapon control are accomplished through algorithms that simulate AMRAAM, SIDEWINDER, and SPARROW missile flyouts, and guns. These routines, which were acquired from Perceptronics Incorporated, have been verified and validated by the Air Force and the Navy on ACMR ranges.

An extensive early warning and ground controlled intercept simulation or Electronic Order of Battle (EOB) was derived from a Northrop analytical non-real-time ground-based air defense simulation called FLEXNET. EOB provides a representation of the elements, functions, and interfaces that exist in an actual air defense system. An approximation of the behavior of a specific defense system is achieved by assigning appropriate values to key parameters such as processing update rates and descriptors representing capabilities of various system elements. EOB is comprised of

- o a single command/control/communication (C^3) center handling weapon allocation and control;
- o a network of early warning (EW) radar sites;
- o a network of surface-to-air missile (SAM) sites;
- o a Fire Control Radar (FCR);
- o ground control intercept (GCI) stations with air intercept vectoring;
- o simulated SU/AWACS and computer controlled aircraft.

This model can be assigned to either the Red or Blue teams. SAM and FCR sites are displayed on the "God's-Eye-View". If, for example, EOB is assigned to the Red team, Blue members will see on their avionics displays occasional radar strobes from the ground-based radars' attempt to establish a track. If a target track is made, the C^3 logic will assign either a SAM site or an airborne interceptor to attack the threat. Blue avionics displays will indicate a SAM site's radar lockon and missile launch, and will provide information on the direction of the inbound missile. SAM altitude exclusion zones can be established to protect friendly air interceptors that are directed to airborne targets from being fired upon. These interceptors can also receive steering cues, shown on the MICS display, that are transmitted from simulated ground control intercept (GCI) stations directing the pilots along an intercept course with the threat aircraft.

Computer controlled aircraft are currently being used to simulate certain interactive participants for some missions. These aircraft have been configured to simulate such varying missions as a Red or Blue AWACS, a Red fighter with limited detection and missile firing capability, or a Blue striker penetrating hostile territory to bomb a target. An effort is underway to integrate the TAC BRAWLER pilot model to enhance the ability of these computer driven participants. As mentioned previously, to augment the number of participants available in the simulation, regeneration of aircraft is possible after a "kill" occurs. The test director selects, from one of the pages on his simulation control display, one of several predefined points in the battle area where he wishes to regenerate a particular "dead" player. A fresh compliment of fuel and weapons is provided to that pilot. This regeneration feature can be used at any time, whether or not a player is dead.

Simulation Initialization and Monitor

ISSL has developed an efficient and flexible simulation initialization function that modifies the load module data base variables and constants to satisfy the simulation run scenario. With menu driven software, these values can be modified quickly between simulation trials causing no delay. The data flow diagram in Figure 8 describes this process. Aircraft types can be predefined in terms of such parameters as airframe/performance vehicle, avionics suite, RCS and IR signature pattern, weapons load, and team and member number. However, any data value can be examined individually and modified. A variety of desired scenarios can be custom built from the various initialization parameters and stored as data files for later use. These parameters include aircraft start geometries (latitude, longitude, and altitude locations in inertial space), force mixes, combat air patrol (CAP) point and waypoint locations, and weather effects on sensors and missiles.

The EOB model is initialized from trial to trial in a similar fashion. This menu driven software initialization routine allows quick modification of parameters in order to represent expected EOB capabilities of either team. For example, SAM and FCR site types and laydowns are stored for a variety of scenarios and can be called up between trials. Other EOB initialization features include the selection of RCS patterns at different polarizations, frequencies, and decibel gains that will simulate the signature of threat aircraft as seen by the ground-based radars.

As part of the simulation monitoring function, data collection and reduction is available in many forms. Real-time event-based data is collected for significant events occurring during the course of a trial. Time-based collection of aircraft state variables, avionics modes, sensor positioning, and other pertinent information is used to playback the engagement in non-real-time. In this way pilot decisions and strategies can be evaluated. Questionnaires concerning human factors decisions are answered by the pilots in real-time at the MICS using the touchscreen to input answers. Quick-look data displayed on a CRT terminal can assess the outcome of the battle at a glance indicating who died and how. Data is stored on the hard disk for later print-out or for transfer to magnetic tape for data analysis at another location. Examples of the type of output derived from some these multiple-engagement simulations is shown in Figure 9.

Extensive data collection and reduction capabilities allow large quantities of results to be obtained. Those results permit the evaluation and validation of aircraft and system designs before an expensive commitment to hardware is made. In addition, by performing integration and checkout of point designs early in the aircraft system development cycle, costly mistakes can be avoided.

Simulation Real-Time Execution

From Figure 10, it can be seen that the simulation real-time executive software manages the load module scheduling. It also provides the interfacing between the domes, MICS, display graphics, and data recording hardware. Much of the system software for schedul-

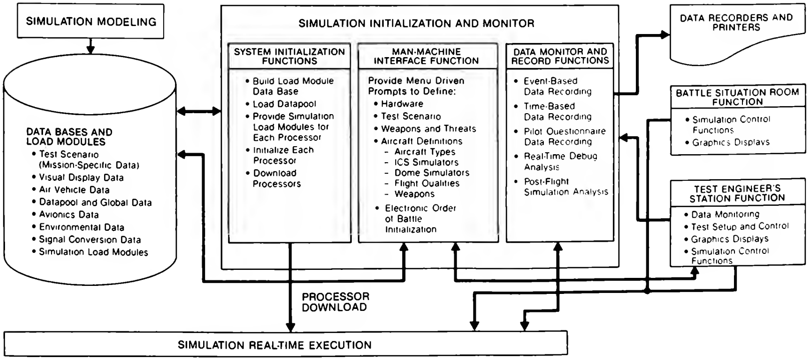

SIMULATION MODELING
DATA BASES AND LOAD MODULES
• Test Scenario (Mission-Specific Data)
• Visual Display Data
• Air Vehicle Data
• Datapool and Global Data
• Avionics Data
• Environmental Data
• Signal Conversion Data
• Simulation Load Modules
SIMULATION INITIALIZATION AND MONITOR
SYSTEM INITIALIZATION FUNCTIONS
• Build Load Module Data Base
• Load Datapool
• Provide Simulation Load Modules for Each Processor
• Initialize Each Processor
• Download Processors
MAN-MACHINE INTERFACE FUNCTION
Provide Menu Driven Prompts to Define:
• Hardware
• Test Scenario
• Weapons and Threats
• Aircraft Definitions
- Aircraft Types
- ICS Simulators
- Dome Simulators
- Flight Qualities
- Weapons
• Electronic Order of Battle Initialization
DATA MONITOR AND RECORD FUNCTIONS
• Event-Based Data Recording
• Time-Based Data Recording
• Pilot Questionnaire Data Recording
• Real-Time Debug Analysis
• Post-Flight Simulation Analysis
DATA RECORDERS AND PRINTERS
BATTLE SITUATION ROOM FUNCTION
• Simulation Control Functions
• Graphics Displays
TEST ENGINEER'S STATION FUNCTION
• Data Monitoring
• Test Setup and Control
• Graphics Displays
• Simulation Control Functions
PROCESSOR DOWNLOAD
SIMULATION REAL-TIME EXECUTION

Figure 8. Simulation Initialization and Monitor

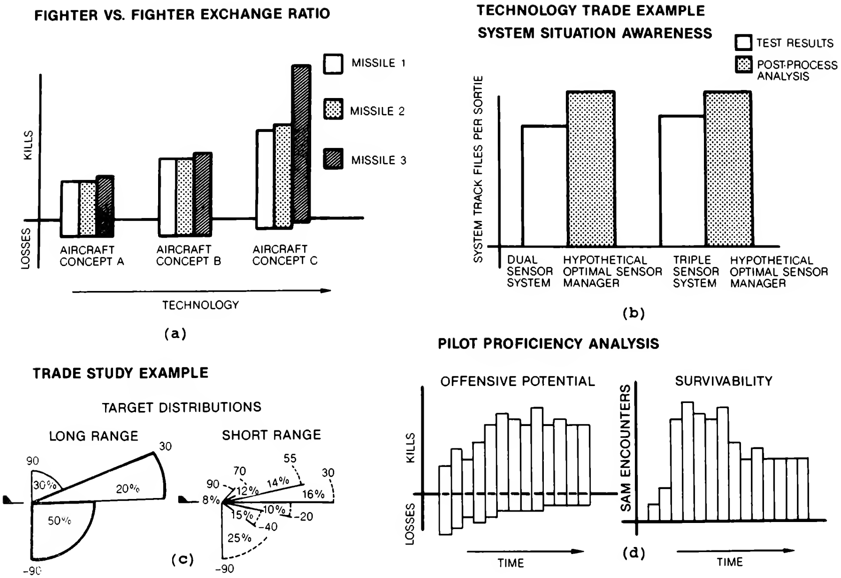


Figure 9. Typical Simulation Results

ing the real-time tasks, downloading of software modules, sending messages, and so forth is developed in-house to better accommodate the unique hardware and software schemes used in the laboratory. System library routines supply the interface between the simulation hardware network and the multitude of I/O devices (digital-to-analog or analog-to-digital converters and real-time peripherals) to transform discrete signals. This type of programmable software switching between digital and electrical lines permits the rapid interchangeability of cockpits and the reprogramming of MICS joystick and touchscreen switchology to meet any project's needs.

SIMULATION PROCESSORS

The processing power behind the real-time simulation incorporates four coupled Gould Concept 32/9780 processors as the primary computing device. The Gould processors control the overall simulation, including the domed simulators and the MICS. The Gould Concept 32/9780s are 32-bit word minicomputers containing a central processing unit (CPU) and a closely coupled multiprocessor internal processing unit (IPU). The IPU does not perform any I/O or interrupt functions, but executes any other instruction type in parallel with the CPU. Both the CPU and the IPU have 32 kilobytes (kb) of cache memory, 256 kb of shadow memory. The two processors share four megabytes (mb) of additional memory. The maximum throughput rate of the I/O system is 26.7 mb per second. The maximum instruction execution is rated at 10 million instructions per second (mips) per Gould 32/9780. All software is run in a 50 millisecond frame time and is primarily developed in the FORTRAN language. Two of the four Gould 9780s communicate with each other through one megabyte of shared common memory. In addition, the four systems are tied into one megabyte of facility shared memory.

In the current configuration, one pair of Gould 32/9780s does the avionics and airframe processing for the Dome 3 simulator, the simulation control displays for the TES and TDS, the Compuscene III interface, the data reduction, the electronic order of battle, the missile flyout and scoring algorithms, and the IR sensor modeling. The other pair of primary 9780s computes the displays and avionics for the seven MICS, the "God's-Eye-View" and the parametric data displays, and controls the audio between participants. When Dome 1 is used in the large air combat simulations, two addi-

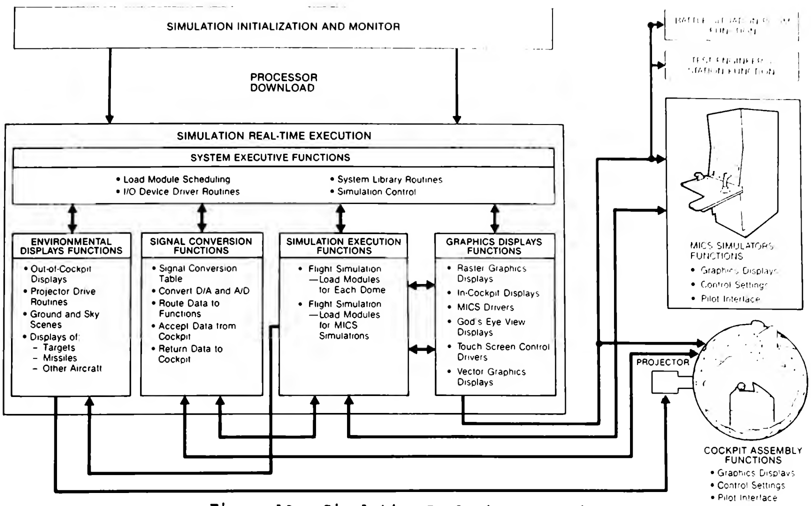

SIMULATION INITIALIZATION AND MONITOR
PROCESSOR DOWNLOAD
SIMULATION REAL-TIME EXECUTION
SYSTEM EXECUTIVE FUNCTIONS
• Load Module Scheduling
• I/O Device Driver Routines
• System Library Routines
• Simulation Control
ENVIRONMENTAL DISPLAYS FUNCTIONS
• Out-of-Cockpit Displays
• Projector Drive Routines
• Ground and Sky Scenes
• Displays of:
- Targets
- Missiles
- Other Aircraft
SIGNAL CONVERSION FUNCTIONS
• Signal Conversion Table
• Convert D/A and A/D
• Route Data to Functions
• Accept Data from Cockpit
• Return Data to Cockpit
SIMULATION EXECUTION FUNCTIONS
• Flight Simulation —Load Modules for Each Dome
• Flight Simulation —Load Modules for MICS Simulations
GRAPHICS DISPLAYS FUNCTIONS
• Raster Graphics Displays
• In-Cockpit Displays
• MICS Drivers
• God's Eye View Displays
• Touch Screen Control Drivers
• Vector Graphics Displays
BATTLE SITUATION DISPLAY FUNCTION
TEST ENGINEER'S STATION FUNCTION
MICS SIMULATORS FUNCTIONS
• Graphics Displays
• Control Settings
• Pilot Interface
PROJECTOR
COCKPIT ASSEMBLY FUNCTIONS
• Graphics Displays
• Control Settings
• Pilot Interface

Figure 10. Simulation Real-Time Execution

tional Gould 9780s are interfaced into the simulation via a Gould-to-Gould High Speed Data (HSD) link. The HSD link is used in this case because of the restriction on large distances in the shared memory between computers. The Dome 1 9780s host that simulator's avionics, airframe, and the interface to the Compuscene IV. Software routines and data pertinent to all participants reside within the primary set of four Goulds. The philosophy here is to avoid running duplicate software, and thus all datapool or global variables are passed over the link.

The vehicle dynamics, equations of motion, and relative geometry are computed in a Floating Point Systems' 5320 array processor hosted to each pair of Gould 9780s. The maximum instruction execution rate on this processor is 20 mips.

As mentioned earlier, the background scene generated by the General Electric Compuscene III system for Dome 3 is driven by two Gould 32/9750s. The Gould Concept 32/9750 is identical to the 32/9780, except there is only one central processing unit (CPU) and no parallel internal processing unit (IPU). These two machines are interfaced to one of the primary set of four Goulds through an HSD inter-bus link (IBL). The Compuscene IV (driven by one 32/9780) is connected to the Dome 1 pair of Gould 32/9780s through the same type of HSD IBL.

The processors driving the color raster MICS, simulation control, "God's-Eye-View", and parametric displays are provided by ADAGE 3000 graphics. The cockpit head-up and head-down displays in both domes can be generated with ADAGE 4135 monochromatic stroke graphics. Currently, color raster head-down displays are being integrated into an advanced cockpit for Dome 3. The Silicon Graphics 4D/70GT is the graphics processor being used to generate these displays.

FUTURE ENHANCEMENTS

Large air combat simulations currently use six Gould 9780s and three FPS 5320 array processors to model nine aircraft, their avionics systems, and the threat environment. This computation effort does not include the processors used to generate the two dome background scenes. Some simulations have saturated this computing capacity due to the complexity of the models required and the number of aircraft involved. The Adage 3000 color raster graphics processor requires a certain amount of Gould processing for display formatting. Hence, when all seven MICS are used along with the "God's-Eye-View", parametric, and simulation control displays, this puts an added computational burden on the host Gould machines. Any expansion to add more participants is not possible with this configuration.

The following enhancements are currently being implemented to accommodate an expansion and enhance computing capacity:

1. Each existing MICS will be front-ended with a Gould Concept 67 Micro-Sel computer which contains a CPU and an IPU. All the avionics, sensor models, and controls and display formatting that is currently done on the host Gould computer will be moved to each individual station's processor.

2. Additional MICS will be added in the near future to meet expanding project requirements. The graphics processor of each of these new stations will be the Silicon Graphics 4D/70GT. Each new MICS will be front-ended with a Gould Micro-Sel. The current configuration, plus the new enhancements is shown in Figure 11. The new stations will mainly be used as Blue participants which tend to have more complex displays. This approach will ensure compatibility with the Silicon Graphics displays generated for the Dome 3 cockpit, also a Blue participant. In order to maintain flexibility for the many programs that are using the facility, these new stations can be Red if a project desires. The video outputs of all the stations will be common (1024 lines) so that one large video distribution system is all that is required to display repeaters at the TDS and TES and record the desired channels.

3. The Red and Blue teams will be given separate brief/debriefing facilities, seen in Figure 12. The MICS will be isolated from one another by locating each station within a small room to avoid intra-aircraft communications from being overheard. The emphasis on flexibility is carried over here by the ability to isolate Red and Blue players for any force mix required using a movable partition. Current plans are to have eleven MICS operational in the near future, though the facility is designed to house sixteen stations for a total number of participants of eighteen.

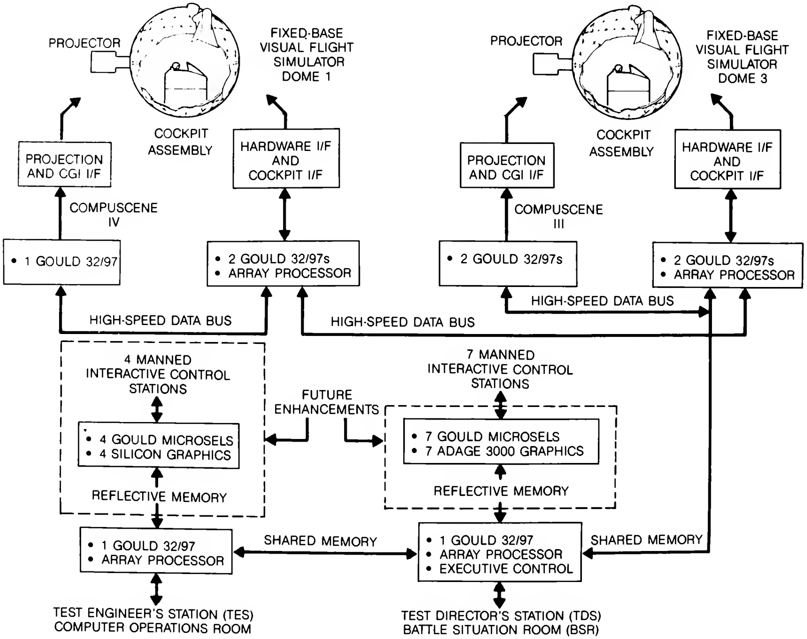


Figure 11. ISSL Current and Planned Hardware Configuration

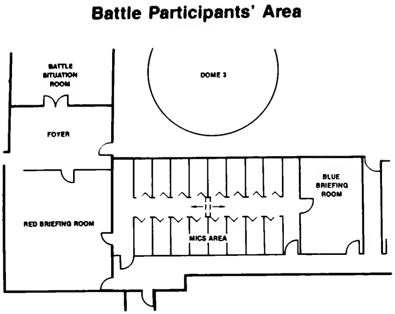


Figure 12. Battle Participants' Area

4. The cockpit assemblies in both domes are to be interfaced with Computer Products Incorporated's Advanced Simulator Linkage System (ASLS). Standalone, this system is run from an IBM PC and performs intelligent I/O to interface any cockpit assembly with Gould, VAX, Harris, and other computer systems. The design and check-out of analog, discrete, and custom interfaces between switches and displays of a cockpit assembly can be done at remote sites before integration and installation into the ISSL domes through a high speed data (HSD) link. The advantage of the ASLS is its remote processor's ability to perform internal conversions of analog, digital, discrete, and ASCII signals, and conduct diagnostics using vendor supplied maintenance software.

ACKNOWLEDGMENTS

I would like to extend my sincere appreciation to those individuals, too numerous to name, who provided ideas and assisted in the proof-reading of this paper. I especially applaud those persons, past and present, who have contributed in all aspects to the success of the multiple-engagement simulation with their ingenuity, hard work, and dedication.

REFERENCES

1. Perkins, J.A., Passmore, H., "Advanced Medium Range Air-To-Air Missile (AMRAAM) Manned AIr Combat Simulation (MACS) Operational Utility Evaluation (OUE)," McDonnell Douglas Corporation, Final Report For Contract F29601-80-C-0044, 31 August 1982.

2. User Operating Manual For Air Combat Maneuvering Range AN/USQ-T2 (U), Cubic Corporation, Contract Number N00019-71-C-0429, 15 August 1974.

88-4600-CP

OPERATIONAL TEST AND EVALUATION THROUGH COMBINED ARMS TEAM COMBAT SIMULATION*

88-4602-CP

R.J. Blizek
B.D. Shipley Jr., PhD

McDonnell Douglas Helicopter Company
Mesa, Arizona 85202

Background

In 1986, Army aviation was facing the prospects of follow-on testing to resolve difficulties with results from the OH-58D Army Helicopter Improvement Program (AHIP) OT-II evaluation. The available options confronted the Army with a dilemma. On the one hand, critics of the scout helicopter concept were threatening to kill the program because the available empirical data did not adequately justify the scout mission requirements. On the other hand, anticipated cost and schedule requirements for additional tests did not appear to be readily supportable within existing budget and major test program capabilities.

A proposed concept, TEACS (Test and Evaluation/Analysis through Combat Simulation), originally submitted to the Army by McDonnell Douglas Aircraft (McAir) in 1985, had suggested that manned simulation could be used to reduce test preparation costs and improve test results. The concept paper had been a result of Army comments about a previous Air Force rental of the McAir multiple player facility to perform an Operational Utility Evaluation of the Advanced Medium Range Air-to-Air Missile (AMRAAM). After reviewing the concept, the Army concluded that the status of existing industry facilities was not sufficient to serve immediate needs and proceeded to conduct phase I of the Army Aerial Scout Test (AAST-I) as a conventional range test.

Since then, several technology advances in critical areas has prompted the Army to revisit the application of manned simulation to improve some aspects of Operation Test & Evaluation. To determine what assets would be required to support Army needs and to determine whether those requirements could be realized with existing simulation technology, is the focus of this paper.

The TEACS Concept

In theory, a multiple player manned simulation capability should provide the Army with cost and performance advantages complementing the analytic models and test ranges currently used to evaluate the mission effectiveness of a weapon system. Figure 1 is a Venn diagram that illustrates the complementary relationship of manned simulation, analytic models and test ranges as methods of estimating performance effectiveness under combat conditions. The region enclosed by the heavy line represents real combat conditions. The heavy line signifies the fact that effects of combat conditions on performance are being estimated, not observed directly.

Three main points are apparent from inspection of Fig. 1. The first point is that a single method does not permit an empirical discrimination between valid and spurious results. Part of each method is located both inside and outside the region enclosed by the heavy line. The part inside represents valid estimates of the effect of combat conditions on performance; the part outside represents results due to spurious effects.

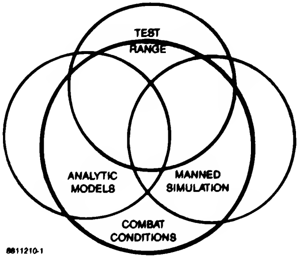


Fig. 1 Complementary relationships of manned simulation with existing methods and combat conditions.

However, since it is not possible to systematically observe the effects of combat conditions directly, there is no empirical basis for determining which results of a given single method are valid and which are spurious. Instead the discrimination must rely on methods of logical analysis and procedural rigor.

The second point is that multiple methods can be used to establish an empirical basis for improving the discrimination between valid and spurious results. Correlation of results among the methods increases confidence that estimates from shared conditions are valid. The triple intersection suggests that adding manned simulation as a third method should increase confidence that some results are valid. Whether or not those results will also represent critical information needed to resolve major decisions remains to be determined. As the third and final point, some parts of the region inside the heavy line are not covered by any of the method areas. In other words, some effects of combat conditions on performance cannot be estimated with any of the methods.

What are some of the expected benefits? The TEACS program seeks a manned simulation system with a capability that will (1) reduce costs through more efficient preparation for range testing and (2) avoid costs through an expanded analytical data base that

*For presentation at the AIAA Flight Simulation Technologies Conference, September 1988.

complements the outcomes of range testing. A comparison with range testing methods suggests that the major strengths of manned simulation would be flexible, highly controlled conditions of performance, and the capability to examine issues that are not permitted on the range due to safety constraints. In addition, simulation offers lower costs per hour of operation as well as greater yield of information per test event.

The Army can expect significant benefits by expanding the scope of testing to include nap-of-the-earth operations that greatly increase safety hazards, particularly night operations. The recently completed AAST-I offers three examples. First, the aircrews were not allowed to operate below treetop level at night as they would in actual combat. This biased the estimates of survivability against the aircraft since it increased the likelihood of detection by the threat air defense. Second, active air-to-air engagements were not allowed. As a result, the requirement to determine the effects of air-to-air encounters on the capability of the team to conduct the primary mission still remains unanswered. Third, the lack of effects from active artillery fire does not task the aircrew to adjust fires and thus underestimates crew workload effects during a critical period of an active engagement.

What are the expected shortfalls? The major weakness of simulation is that state-of-the-art methods of producing conditions of performance in some areas, e.g., visual scene, still cannot provide the richness of detail and extent of information that would be encountered under conditions of performance on the range. Relative to analytic simulation models, e.g., war games, manned simulation offers active participation of system operators using high fidelity models of actual equipment against representatives of real opponents. These features could then provide data that would qualify critical assumptions made in the analytic models, e.g., about effects of human performance. The main weakness is that manned simulation cannot represent large force engagements.

Conceptualizing a TEACS simulator

Fig. 2 and Table 1 present the levels of participation as a top down hierarchy proceeding from control at the top to primary test participants at the bottom. The importance of each simulation level relative to quality of test performance is indicated by the print size of the test functions; larger is more important.

The three functions at the top, control and data acquisition, are not separated between test participation and simulation capability because they were found to represent equivalent technologies in both cases. They are depicted as overarching capabilities that have access to all lower levels.

Below the control and data acquisition functions are the force components. The force components are separated into red and blue teams of parallel capabilities. The blue force, which is typically the test subjects, will all always require full mission simulators that provide test items and the other vehicles of the test unit. The red force will only require a full mission simulator when a test scenario calls for sustained high level performance against a blue test item or test unit member; e.g., aggressive helicopter versus helicopter air-to-air engagements.

In the middle area of the figure, between the red and blue capabilities, the "environment" capability which is not associated with any T&E participant roles. In this illustration, the environment represents functions needed to integrate and support the performance of the red and blue forces and their capabilities. At minimum, the environment includes some means of exchanging data that defines the effects of interactive play between the two forces. It may also include infrequently used functions that are common to many players or functions that operate on information obtained from several players. Some examples are weapon models, line-of-sight determination, and real time damage assessment.

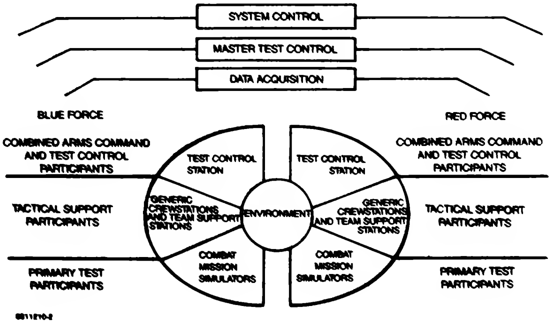

SYSTEM CONTROL
MASTER TEST CONTROL
DATA ACQUISITION
BLUE FORCE
RED FORCE
COMBINED ARMS COMMAND AND TEST CONTROL PARTICIPANTS
COMBINED ARMS COMMAND AND TEST CONTROL PARTICIPANTS
TEST CONTROL STATION
TEST CONTROL STATION
TACTICAL SUPPORT PARTICIPANTS
GENERIC CREWSTATIONS AND TEAM SUPPORT STATIONS
ENVIRONMENT
GENERIC CREWSTATIONS AND TEAM SUPPORT STATIONS
TACTICAL SUPPORT PARTICIPANTS
PRIMARY TEST PARTICIPANTS
COMBAT MISSION SIMULATORS
COMBAT MISSION SIMULATORS
PRIMARY TEST PARTICIPANTS
88112 10-2

Fig. 2 Correspondence betweenT&E roles and simulation capabilities that represent equivalent functions.

Table 1 Depicts the correspondence between T&E roles and simulation capabilities that represent equivalent function.

Participant Role	T&E Function
System control	Monitor and evaluate overall status of system operations.
Test control	Monitor, evaluate and manipulate performance of supporting participants to create situations that conform to requirements of trial objectives.
Data acquisition	Monitor, evaluate and report operational status of real time test instrumentation systems.
Command and Control	Service all communications requests in support of tactical play and test control manipulations; separated by red and blue forces.
Red forces	Provide tactically viable threat maneuvers against blue forces.
Blue force support	Provide combined arms tactical maneuver and interactions with blue unit under test.
Blue force test unit	Provide trial performance using items under examination in trial objectives.

A Baseline System Configuration

The breakout of simulation functional capabilities shown in Fig. 2 can be distributed over component technologies with a variety of subsystem configurations. Fig. 3 presents a subsystem level elaboration showing a flexible configuration of the full mission simulator subsystems that will support tests to evaluate modernization of the existing fleet of scout and attack helicopters. The number of full mission simulator subsystems reflects the organization of the light platoon of the attack helicopter company - a scout and two attack helicopters.

The main features of Fig. 3 are the subsystems of the full mission simulator complex, the means of providing the supporting or auxiliary players including a remote interface with SIMNET/AIRNET, and the relocation and subdivision of the test control function. Each full mission simulator complex is shown with an image generator to produce the visual scene, a dome and projector to display the visual scene, a crewstation, and a computer to provide vehicle dynamics. This breakout does not accurately reflect the subsystem structure used in the technology analysis. But it does highlight some important features of the simulator system.

The visual subsystem includes the image generator and the display media as component technologies. The three arrangements analyzed (dome and projector, collimated, and helmet mounted displays) emphasize the critical relationship between visual display technology, the crewstation subsystem, and the flexible configuration requirement to support different user test vehicles.

The local supporting tactical player capability (labeled "AUX PLAYERS") is indicated by an undifferentiated (red versus blue), indefinite number of relatively inexpensive visual displays. SIMNET/ AIRNET is shown as a remote implementation of this type of technology.

The test control subsystem has been relocated from the top area of Figure 2 to the bottom area of Fig. 3 primarily to suggest possible physical separation of system and test control subsystems and actual separation of red and blue subordinate components. It is important to separate the red and blue forces as well as

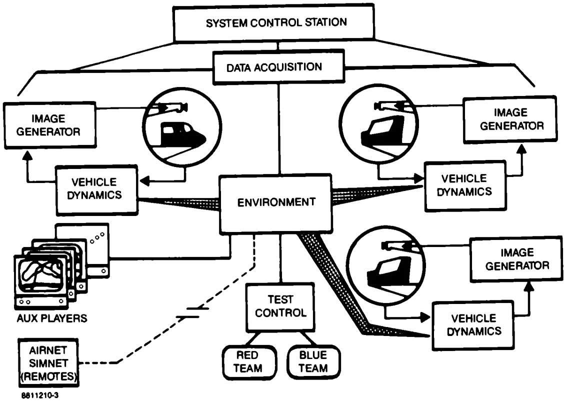


Fig. 3 Baseline configuration concept.

the subordinate test support elements to minimize "intelligence" leaks that might compromise the execution of test trials.

An implicit feature of Fig. 3 is the method of intcgrating systems through the environment computer. The allocation of functions among subsystems and components, and choices among options for integration architectures are two major sources of variation in technology requirements. That is, different combinations will have different effects on overall system performance, development risks, and costs. It follows that concerns about integration must be a predominant factor in the definition and evaluation of component technologies.

Simulation Capabilities Versus Requirements

By comparing current state of the art simulation technologies with operational test capabilities, an assessment of how well simulation technology can accommodate test requirements can be made. A tabular format is used to present the essential information. Each table lists a subsystem (and component where appropriate) and one or more critical requirements for that subsystem/component. Then, for each requirement a state-of-the-art simulation capability is given with an estimated percent of that requirement provided by the capability.

Finally, there is a recommended approach in cases of significant limitations or opportunities for improvement of performance. Most of the information is concentrated in the visual and crewstation subsystem sections because they are most critical to performance of comparison items and quality of test results.

Visual Subsystem

The visual subsystem summary is given in Table 2. The critical tactical performance requirements involves detection range, object density, display resolution, and image contrast as key requirement issues. The following summary assumes a dome with a projector as the display method because of the greater relative importance of wide Field of view (Hughes and Brown 1983). The two critical requirements of object range and density are presented together because they effect each other directly (Blizek 87).

The tactical objective is to effect timely, accurate visual search of the local area for presence of significant objects. A "significant object" is one that must be given extra time for special examination because it may be an enemy vehicle that is either an immediate threat, a potential target, friendly, or a source of useful information about the general situation. Groups of trees or buildings must also be treated as significant objects because they provide locations where the enemy might conceal vehicles. To provide satisfactory support of the visual search task, the visual subsystem capability must therefore provide enough objects to produce the desired visual search patterns.

In addition, these objects must have enough detail to enable discriminations to be made among threat and non-threat objects in representative periods of time and over distances that should be of greatest concern during combat operations.

Detection Range Versus Object Density

Out-the-Window

The out-the-window 3-5 kilometer range is nominal, but not a close approximation of the maximum possible in the real world. The AAST scenarios called for 5-7 km to be varied according to the situation. However, 3-5 km is a reasonable standard based on line of sight probabilities for low level flight operations (Fig. 4).

Task definition is important to an evaluation of the capability. Given that visual search activity is the critical behavior and that it starts as quickly as objects appear (are detectable) in the visual scene, the 2-3 km value is defensible. It is not if the important behavior is the actual discrimination of significant from non-significant objects (say "recognition" rather than "detection"). The argument is actually over task definition rather than image generator capability. The important point is that both behaviors can occur within this range since 2 km was about the median of the AAST-I detection for the OH-58C.

Table 2 Visual system requirements summary.

Subsystem/ Component	Requirements	Simulation Capability	Estimated Percent of Requirement		Recommended Approach
Image Generator	Detection Range:				
	3-5 km (OTV)	2-3 km possible	60		
	10-15 km (sensor)	7 km	50		Special IG
	Scene densities	Sparse	40		Special features
Display	Very high resolution	High resolution	70		Done
		Very high resolution	100		HMD - option
			AV	EX	
		High contrast ratio	10	1	Done
		Real world >1000	50	5	HMD
		Average 1:100			

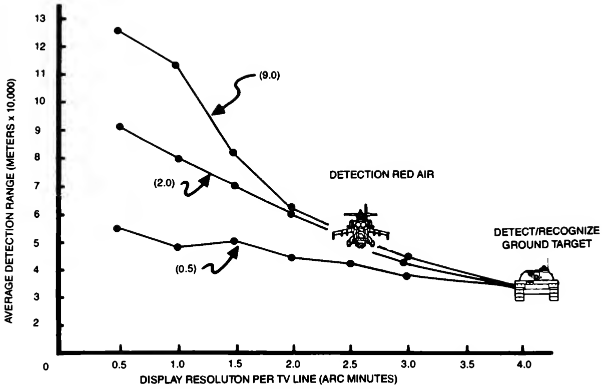


Fig. 4 Average detection range as a function of display resolution and target (head-on) scenario.

Although image generator capability is sufficient, it is at the lower limit and greater detection distances are desirable. In fact, greater ranges are achievable for situations that do not demand high levels of object density and detail; i.e., a desert environment.

For TEACS, the visual scene must have enough clusters of objects to create situations that cause the aviator to engage in effective visual search behaviors. The image generator cannot accomplish this object density at long visibility ranges. Therefore, greater scene density may be preferred over greater visibility range.

The conclusion reached is that as a constant over trials and test units, the effects of the visibility range and object density limitations should not affect comparisons of systems being tested. Comparison of trends should be valid, although comparisons of absolute values would not be.

This will allow for direct comparison of weapons systems relative to a common data base and evaluation criteria which is one of the main points behind the TEACS concept.

Sensors

In one respect, the sensor image offers an advantage for the image generator because they typically use a narrow field of view. The narrow field of view concentration of objects in smaller areas for much greater distances. However, advanced sensors will require "visibility" ranges that are about twice the available image generator capability. Since improvement of sensor performance is a major objective of fleet modernization research, the industry need is to provide a capability that will satisfy the longer visibility range.

The current available capability covers most sensors now in service. The simultaneous presentation of out-the-window and sensor scenes with a single image generator is a problem and will become more serious for advanced sensors. By employing two image generators, the major area of concern is the need to establish high correlation of the positions of objects between the out-the-window and sensor IG data bases. High correlation of culture positions and terrain geometries is essential for accurate assessment of tactical performance across systems.

Image Generator Conclusions. Remarkable improvements in visual scene capability (image quality, texturing, etc.) have been realized in the newer image generators, at least to the point that support of high quality performance of OT&E requirements is now feasible. However, even more improvements are needed. The visual system industry recognizes this need, and is making substantial investments to provide better performance. The timing of breakthroughs is a judgment call, but 3 to 5 years seems reasonable for a major increase in capability. Meantime, the prospects are for modest increases of performance and lower costs as the current technology matures.

Displays

Resolution

From a practical perspective, resolution of a display is determined by the size of the smallest area in a scene with values (e.g. color, brightness, etc.) that can be controlled directly by the image generator. The resolution of the human eye is regarded as the standard. It is capable of resolving objects that cover about 1.3 arc minutes of visual angle.Existing area-of-interest projectors provide resolution of about 2.3 arc minutes and helmet mounted displays offer about 1.5 arc minutes.

Although desirable as a limiting goal, the eyeball standard seems unreasonable given the fact that current visibility range limitations of image generators are not sufficient to fully utilize the resolution capability of existing area-of-interest projectors. The helmet mounted display technology was not recommended because of current image generator limitations; but more importantly, this technology is not compatible with crewstation components like the Apache ORT or with the need to conduct tests that include the use of night vision goggles and MOPP gear.

Therefore, one may rightly conclude that a major breakthrough in image generator technology is needed to realize the full potential of the helmet mounted display technology. The recommended approach will accommodate the mixed use of area-of-interest and HMD displays.

Contrast

Contrast is a ratio (several measures are recognized) of the lightest to the darkest area in a visual scene. It is an important, but still controversial, display value over which there is much research and debate. It is important for all types of displays, e.g., heads-up and multifunction displays of modern high performance aircraft. There are two contrast effects (local and global) to consider. Global contrast is a function of the most extreme values available in the visual image. It expresses the range over which the image generator is able to manipulate local contrast effects. Local contrast is the relative difference in values between two adjacent areas. Variation in local contrast is needed to reveal differences among textural details of similar color in a visual scene and is a standard display capability.

In practical terms, the maximum global contrast in a typical sunlit scene is due to extremes of brightness that can never be achieved with any real time simulation display technology. For example, the glint of the sun from a window pane in a door swinging in the breeze will attract attention instantly. In fact, such cues would not be desirable in an operational test scenario on the test range because they create conditions that are not standard from one test player to the next.

Contrast in a dome is affected by the amount of light needed to cover the size of the display area. The greater the size of this area, the lower the contrast due to the diffuse reflections of the light throughout the dome. In fact, the typical multiple projector, large field of view domes used in the fixed wing industry (to provide displays for simulators used as engineering design tools) only have contrast ratios of about 2 to 1.

The single head-tracked projector approach offers contrasts of about 10 to 1, and experiments with a new high gain paint at MDHC shows a value of about 20 to 1. The helmet mounted display offers the best available contrast ratio, and at about 50 to 1, it seems to be approaching the limit of that technology.

The existing technologies appear to be at their limits of contrast performance. New, higher resolution methods of light emitting displays with full color must be developed to produce substantial gains. The prospects of achieving these features seem unlikely for sometime due to the limited marketability (beyond simulation) of such products.

Visual Subsystem Conclusions. In terms of tactical performance for TEACS, the main conclusion is that the existing technology will permit the generation and display of useful visual scenes. The most desired value is improved image generator capacity to provide greater object density. Some increased capacity is currently available, at a substantial cost, and may not yield that much overall increase in quality of performance due to nonlinearities. That is, doubling the capacity would not produce twice as many trees nor a greatly extend visibility range. Improvements of the image generator are needed to realize the resolution capability of existing display technologies. Increasing contrast is also a desirable trend for newer display technologies.

Crewstation

The crewstation summary is provided in Table 3. From the OT&E perspective, the conventional intent of a test is to determine the extent to which any new or modified combat system meets the required operational capability for which it was designed. The tactical objective is engage the crew in operating the actual equipment or the best available approximation that simulation technology can provide.

The crewstation subsystem must enable the tester to examine the performance of the total man/machine system operating with other members of the light helicopter platoon and the blue combined arms force against a credible red opponent force. The crewstation of the vehicle under test must have very high engineering fidelity of the component functions and the operator interfaces. Lack of such fidelity is likely to result in refusal of decision makers to accept the results.

Two methods are used to provide the required capability. An obvious method, often called stimulation, is to employ the actual system equipment in the simulation crewstation. The option, called emulation here, but also referred to as "simulation," is to use models of the actual equipment. There are reciprocal trade-offs with either approach.

Regardless of its intuitive appeal, the stimulation approach may require considerable effort to integrate the actual equipment so that it will operate properly in a simulation environment. The equipment often operates at different frequencies and was not designed for the effects of such simulation operations as "freeze" or data acquisition. Another issue is providing the actual equipment with the necessary signals to cause it to function as it would in the aircraft.

Because crewstation instrumentation/switchology features are not easily accommodated by emulation capabilities, stimulation often becomes necessary. The major benefit of the stimulation approach is assurance that the functions of the actual equipment are being examined. Emulation, on the other hand, reduces the difficulties of integration. The emulated equipment models are designed and developed to operate in the simulation environment. On the other hand they require extensive test and validation work to demonstrate that their functions are satisfactory representations of those provided by the actual equipment. This verification process is seldom simple and the effort to accomplish it is too often underestimated.

With recognition of these limitations, either approach, or more often a mixture of the two can provide at least 90% of the required aircraft functionality. Therefore, it is recommended that emulation should be used where possible, stimulation where necessary.

Table 3 Crewstation system requirements summary.

Subsystem/ Component	Requirements	Estimated Simulation Capability	Percent of Requirement	Recommended Approach
Crewstation	Aircraft functionality	Emulation/simulation	90	Emulation
Vehicle Dynamics	Emulate vehicle	Improve FLYRT	80	AH-64 (NASA/ARTI)
		ARMCOP	80	OH-58D (Bell)
		Blade element	90	Air-to-Air

Vehicle Dynamics

The vehicle dynamics subsystem must fully support the dynamics of the simulated vehicles that are involved to accomplish tactical objectives for both air-to-ground and air-to-air missions. The dynamic performance of the simulation must enable the crewmembers to experience vehicle control tasks as nearly as possible like the actual aircraft.

The FLYRT (a warpable disk table lookup model) and ARMCOP (a rigid disk model) models of aircraft vehicle dynamics are currently available, although work is underway to improve the documentation and performance of FLYRT. These models will operate with existing simulation computer technology.

The blade element approach will make it possible to provide high quality performance of the aircraft at the extremes of the flight envelope; a feature that is lacking in the FLYRT and ARMCOP models. Blade element models are used now in research with batch mode methods of processing. The development work needed to adapt them to real time processing requirements has only recently been undertaken.

As with crewstation functionality, vehicle dynamics are intrinsically involved in the performance of the system and deviations from the performance of the actual system are potential sources of contamination in the test results. Even though the industry has typically addressed performance deviation by advances in rotor dynamics modeling, NASA Ames studies have concluded that a number of other factors need to be addressed as well. Such factors include drag and rotor/fuselage interaction, ground and stall effects as well as fin and stabilizer effects. The development of modeling techniques to account for these effects needs further development by the industry (Marsh 88).

An unresolved issue is how the models used to represent vehicle dynamics will be certified by the Army as sufficient to meet performance requirements of the objectives of any given test.

Auxiliary Player

The prediction of combat effectiveness of a candidate weapon system requires that the test provides a viable threat force. In addition, the blue force combined arms team must also be provided. Providing these components in a cost effective manner is the purpose of the auxiliary player subsystem.

A summary of the auxiliary players is given in Table 4. The auxiliary player station must provide all essential means of control, communication, and interaction with the other players and the full mission simulators. The visual display should have the capacity to support medium levels of scene detail as compared with the capability of the full mission simulator. The time delay should should provide vehicle dynamics of either helicopter or fixed wing aircraft in air-to-air maneuvers near the ground; a representative value is 150 milliseconds recommended for full mission simulators. For network time delay, 250 milliseconds without benefit of linear prediction, and a maximum of 500 milliseconds with prediction have been suggested (Malone III, Horowitz, Bunderson and Eulenback, 1988).

An auxiliary player station must provide the means of maneuvering and employing a platoon formation of ground or air vehicles. The maneuvering

Table 4 Auxiliary Player system requirements summary.

	Subsystem/ Component	Requirements	Estimated Simulation Capability	Percent of Requirement	Recommended Approach
LOCAL	Visual Display	Medium resolution	Low resolution	30	Upgrade to medium resolution
	Vehicle Dynamics	Low time delay	Exists	60	Microvax or equal
	Remote	SIMnet integration	Exists	100	Ethernet compatible

feature must provide for movement of the formation from the position of formation lead. Each following member of the formation must have terrain following capability. Avoidance of microterrain is desired.

Employment of the formation involves the capability to take the perspective of any member of the formation. When in deployment mode, the formation control mode will permit maneuvering of any one member from any position to any other position in the data base. During this period of active individual control the remaining members of the formation must be in established, fixed positions, or under the separate point to point movement control of software.

Existing formation control procedures do not provide individual member terrain following and trajectory capability. The needed level of formation control is, however, an extension of existing capability and is within state-of-the-art technology. The need is for development to upgrade the current capability. This is an area of industry investment and is expected to be available within the period of time needed to incorporate it into TEACS.

The current low cost graphics workstations provide about 30% of the estimated requirement. The limitation is the capacity to generate the type of information needed to enable the player to respond effectively to detection and engagement of opponents. The limitation of the vehicle dynamics is time delay. While the current capability is minimally sufficient, increases in number of players will increase the risk of data transmission lags that will progressively degrade the performance capability. To avoid this risk, the approach is to provide sufficient local processing capacity to minimize the contribution of this factor to the overall time delay. Essentially, the need is to provide a system that will support smooth, integrated movements of rapidly moving objects.

Environment Subsystem

The environment subsystem is the source of weapons effects, real-time casualty assessment, and similar tactical service support functions. It is essentially the simulated combat area in which all participants interact.

Environment subsystem ratings are given in Table 5. The requirement is for weapons models that are based on algorithms that provide 3 degrees of freedom. The 3 degree of freedom models exist and are used widely, e.g., test ranges. Recently, NASA/ARTA has found that 3 DOF weapon models were not capable of providing accuracy needed to determine damage assessments based on characteristics of the area of the target for precision guided and point accuracy weapons. The 6 degree of freedom models, however, offer the prospects of increased performance. They are now in use in batch processing applications. In fact, NASA/ARTA has defined the 6 DOF capability as a requirement for the CSRDF improvement effort. On the other hand, 6 DOF models will require the employment of array or parallel processing computer technology. Since the computer hardware technology is now available from industry, conversion of batch mode software to support real-time operations is a lesser risk than before. This capability would provide TEACS with a capability that exceeds the current test range capability.

System and Test Control Subsystems

Real Time Operation

The Blue and Red forces command and control processes of combat operations and communications and control of artillery effects are incorporated into the TEACS scenarios through the provisions of the test control component. Adjustment and employment of artillery are tasks that cannot be performed effectively in the test range environment but can be easily incorporated into the simulator. Provisions for participant briefing and debriefing must be included to cover those dimensions of tactical performance that are part of the pre- and post-mission activities.

At the level of system control, the primary requirement during operation of TEACS is acquisition and selective display of real-time data. Other real time system control functions that must be supported are the monitoring and display of data reflecting the functional status of all subsystems. A comparison of data collection capabilities between simulation and field testing is given in Table 6.

The test control subsystem must have the displays, controls, communications and physical arrangement to permit the test director, the red and blue test control staff, and the red and blue force supporting participants to perform their functions. The capability must provide information displays that will permit the test director to monitor the state of all forces in the unfolding tactical situation. These displays should be configured to enable the test director to examine the tactical situation or weapon system status of any player in either force from the perspective of the player.

Table 5 Environment system requirements summary.

Subsystem/ Component	Requirements	Estimated Simulation Capability	Percent of Requirement	Recommended Approach
Weapons Models	3 DOF 6 DOF desired	Exists Under development	100 100	Array/Parallel processing
Countermeasures	Jamming High energy laser	Exists Emulation	Acceptable	Strobe
NBC	Full MPOO gear	Exists	100	Special heating/cooling

Table 6 Data sources trade-off analysis matrix.

Data Item	RATING		Operational Testing (Hunter-Liggett)	Simulation Technology Available
	OT	SIM		
1. Position/Location		+	Cartesian Coordinates (X/Y) CDEC (RMS) Accuracy within 50 at 3 second intervals CDEC (RMS) and manual data collection	Placement of all components known to within 1 foot resolution at 30 Hz sampling rate. Data collection and recording auto matic. Replay of trial flight and ground paths of short duration (5-10 minutes) are possible.
2. Player/Crew Identification Data includes vehicle type for all players		+	Accuracy 100% – CDEC (RMS) & RTCA sources and manual data collection	Accuracy 100% – Identification of all active participants inherent in the environment. Data collection can be automatic with Players identifications under run time control.
3. Line-of-sight (Exposure Between Players) Entire playing area	+		Continuous update Accuracy <60% – ELOSS – TCATA – TSV – manually collected data – post trial reconciliation 90% helo exposure to target arrays (3000)	CIG Approach: Limited to help exposures to ground targets 360 degrees around A/C Ranges Players: 16 air or ground target arrays. No player intervisibility. 15 Hz update 100% Accuracy External Processor Approach: Possible for all players Ranges Players: 20 possible 40 possible w/technology risk 1-3 sec update 80% Accuracy
4. Through-sight video			Comparable resolution to crew presented video. Recorded to MISC or RS170 video tape format.	Comparable resolution MISC
5. Through-sight video (Ground Threats)			OPERATIONAL TESTING Sufficient resolution to determine target system types	Simulation technology available MISC
6. Cockpit Video			1553 data words incorporated w/video.	Video can be timed tagged with any recorded data for later correlated retrieval.
			Accuracy 100% – Resolution sufficient to determine controls and switches used by aircrew. Also used as motion and down range masking indicators.	Accuracy 100% – Limited ambient light and brightness contrast between display system and cockpit can be overcome with off-the-shelf low-light level CCD cameras. Video can be used in same manner as operation test methods.
7. Reports			Sources are time tagged and are from audio and cockpit video.	Comparable capability
8. CDEC Voice Recording Systems (VRS)		+	Correlated with ATHS and cockpit video all voice by all cockpits (radio/intercom)	Voice recording on time correlated video all voice communications placed on single sound track. Run time control of voice inputs possible. Aircraft, Static, and Munitions noise can be excluded from voice track.
9. Ground-to-Air Detection/ Recognition/Location		+	Azimuth Angle Accuracy (3 degrees) Exclusive and annual collection Post trial reconciliation Accuracy 100% Time Correlation (RMS) Accurate to (3 seconds).	Azimuth Angle Accuracy (1 degree) Run time data collection Accuracy 100% Time correlated (function of LOS Approach) CIG Approach: Correlatable to within 70 ms from event External Approach: Correlatable to within 3 seconds of event.

Table 6 Data sources trade-off analysis matrix (cont).

Data Item	RATING		Operational Testing (Hunter-Liggett)	Simulation Technology Available
	OT	SIM		
10. CDEC RTCA				
- Range		+	< 100s	: ft resolution
- Time (Trigger pull)		+	Accuracy (3 sec.)	Accuracy: 70ms
- Munitions			100% accuracy	100% accuracy
- Method of Engagement Manual, radar controlled			100% accuracy	100% accuracy
- Seeker Status	+		100% accuracy	100% accuracy fidelity of seeker model an issue.
- Critical Illumination	+		Run Time Accuracy 60% Post trial analysis 80%	100% accuracy using CIG illumination control.
- % of Ground Target		+		10% Increments Exposure dynamics limited
			80% Accuracy	10% Increments 100% accuracy
- Ground Target to Firer Aspect Angle		+	Post Trial assessment using TSV 45 degree increments Accuracy 80%	RT data record 1 degree resolution Accuracy 100% Repeatable
- Ground Target to Observer		+	Post Trial Assessment using TSV 45 degree increments Accuracy 80%	Run time data record 1 degree resolution Accuracy 100% Repeatable
- Exposure of Ground Target at missile/ round impact			Post Trial Assessment using TSV Accuracy 100%	Function of LOS approach CIG approach: 100% accurate to 15 M resolution
At Air Target			Accuracy 100%	External Approach: 100% accurate to
- Exposure of Ground Target				3 second resolution

The capability must provide communications and system control interfaces that will permit the test director to manipulate or guide, directly or indirectly, the current actions of any player to effect conformance with the requirements of the trial objectives. The displays must enable the test directorate to assess the state of real time data acquisition and system status on demand.

The test control subsystem capability must provide the red and blue force command and control and artillery participants with the displays and communications interfaces to enable them to interact with and affect the engagement of the members of their respective force.

The conclusion is that the available capability will permit the development and operation of TEACS at levels that exceed the equivalent range capabilities.

Non-Real Time Operations

In the pre-trial phase, the staff of the test directorate must have the capability to configure and verify the functional status of all aspects of a trial scenario, the data acquisition and real time data displays, the auxiliary player stations, and the full mission simulators. The facility must be capable of supporting the briefing of the test directorate and test participants.

In the post-trial phase, the staff of the test directorate must have the capability to debrief the test directorate and test participants, to verify the integrity of test data immediately upon completion of a test trial, and to accomplish a "quick look" assessment of the test results prior to the set-up and execution of the next trial or replication run of a given trial. The capability shall also exist to enable the test support staff to perform statistical analysis of test results by integrating and processing the data from a series of test trials for the purpose of providing answers to test issues.

The test and system control subsystems must be configured with the necessary computer hardware and software processing capability to support non real time data management functions to include data storage, data retrieval, and data manipulation. The system control subsystem must support the operation of selected subsystems or subsystem component functions in non real time mode.

Findings

The results of the study indicated that the concept would meet identified Army operational test requirements. Using the light platoon of an Attack Helicopter Company as the baseline case for analysis, previous operational test plans and reports were reviewed to identify test limitations and tactical performance requirements. The results were used to derive a baseline system definition.

The concept definition was comprised of (1) three visual display systems based on dome technology, (2) a network of eleven auxiliary player workstations configurable in various combinations of supporting air, air defense, and ground maneuver units for red and blue, with provisions for a remote link to SIMNET/AIRNET, (3) a test control center with provisions for the test director and his staff including artillery functions and headquarters command and control communications for both red and blue, and (4) a system control and real time operations center with data gathering and analysis capability. The concept definition and performance requirements were then compared with capabilities of existing simulation technologies to identify limitations and development risks. Dome display technology was chosen over helmet mounted and flat plate collimated displays to provide low cost wide field of view capability while readily accommodating provisions for MOPP gear and crewstations with equipment of vehicles in the existing fleet (e.g., Apache Optical Relay Tube). All required technologies were found to exist and identified limitations are currently being improved by industry or government efforts with results that could contribute to the development of the desired facility within the next 12-18 months. The inability of existing visual image generation technology to simultaneously provide the operator with sufficient density of high quality objects (e.g., targets and trees) and real world detection ranges in the visual scene were analyzed with two main conclusions. First, existing capability is sufficient to provide equivalent task performance by the operator. Second, any bias relative to real world performance would be constant for both red and blue players and comparison of trends, but not absolute differences, among competing systems under test would be valid relative to results of range tests.

An examination of industry average costs for materials and typical labor estimates indicated that the prospective system could be a cost effective solution to the Army's needs for improved testing methods. Finally, a top level comparison of advantages and disadvantages for five implementation options indicated leasing from industry would provide the best value to the government.

industry lease approach. This approach would enable the government to obtain actual cost and performance data to support a definitive cost/benefit analysis without making a substantial capital investment in the most expensive simulation technologies. The government would need to underwrite the cost of test unique items not currently available in the industry.

References

1. Marsh, G., "Modeling Rotor Blades – The Way Forward?" Defense Helicopter World, Volume 7, No. 2: April-May 1988, p42-43.

2. Bellman, B., Smart, D., Bugh, G., "Multiple Sensor Image Presentation," Image IV Conference, 23-26 June 1987.

3. Malone III, H.L., Horowitz, S., Bunderman, J.A., and Eulenback, H., "The Impact of Netword Delay on Two-Ship Air-to-Air Combat Simulation," AIAA Flight Simulation Technologies Conference, New York: American Institute of Aeronautics and Astronautics, August 1987, p55-58.

4. Richardson, P.O., Zittleman, N., and Seale, L.K., "Army Aerial Scout Test," Phase I (AAST) Independent Evaluation Report (SECRET) (IER-FO-1272). Falls Church, VA: U.S. Army Operational Test and Evaluation Agency (USAOTEA), September 1987. (Distribution limited to U.S. Government agencies)

5. U.S. Army Combat Development Experimentation Center, Army Helicopter Improvement Program Operational Test II (SECRET) (AHIP OT-II) (TR-OT-85-744) Falls Church, VA: U.S. Army Operational Test and Evaluation Agency (USAOTEA), September 1985.

6. Blizek, R.J., "Area/Polygon and Z-Buffer Image Generators: Current Capabilities and Future Trends," R.J. Blizek, American Institute of Aeronautics and Astronautics Flight Simulation Technologies Conference, September 1987.

* Study conducted under contract
to TRADOC, U.S. Army

EXPLORATION OF ADVANCED GROUND ATTACK SYSTEMS BY MANNED SIMULATION

88-4603-CP

D. N. Jarrett
Mission Management Department, Royal Aerospace Establishment,
Farnborough, Hampshire, GU14 6TD, England

Abstract

As part of a simulation suite, RAE Farnborough has developed a manned simulation facility for studying the operation of novel avionic and weapon systems during crucial phases of ground attack missions. The simulator is described as a set of functional units. Some of the studies, into terrain avoidance systems and display devices are reviewed. The problems of conducting manned simulation experiments, the impact of simulation limitations, and suggested pre-conditions for extrapolating conclusions to real aircraft operations, are discussed.

[Any views expressed are those of the author and do not necessarily represent those of the Department.]

1. Introduction

We are interested in developing equipment, procedures or tactics which help the pilot of a ground attack aircraft to operate more effectively. Having devised such potential benefits, it would be best to integrate the necessary systems into a front-line aircraft and then sally forth into the nearest convenient conflict to put them to the test.

Lacking the resources and dire necessity to progress in this definitive but haphazard way, we use instead a more considered elimination, starting with theoretical and laboratory studies before entering simulation and flight trials. Simulation offers the advantages of not requiring flight standard equipment; indeed the function of hypothetical systems or weapons can be modelled mathematically. It can also generate the complexities and hazards of conflict more cheaply and safely than genuine flight, and it is easier to obtain the data for objective evaluation.

The topics which are amenable to study by manned flight simulation are intrinsically aircrew-centred, typically involving the arrangement of displays and controls in the cockpit, or assessing human skills. Most concern the pilot of a single-seat aircraft who is required to complete a barrage of demanding tasks at a pace which is forever forced by flying progressively lower and faster to evade enemy defences.

2. The simulation facility

The ground attack simulator is part of a facility which also enables study of air combat and rotary wing applications. The equipment is modular, and can be tailored for specific trials.

Generally, for air-to-ground experiments the pilot is provided with the essential visual conditions of flying at low level and high speed over hilly terrain containing vertical obstacles such as pylons and towers. The full colour external scene is projected by a single window television system having 30 by 40 degrees field of view, fixed in the forward direction. Flight, weapon aiming and steering information is superimposed, as in a head-up display, using an optical combiner. Both are collimated to optical infinity and, together with conventional cockpit instruments and a moving-map display, they give the normal daytime conditions.

The current night-flying equipment fitted to operational aircraft is a fixed forward-looking infra-red camera which supplies an aligned unmagnified image on a television raster HUD. The pilot uses this good quality but restricted view ahead in conjunction with a set of helmet-mounted night-vision goggles, which enable him to look out and around. The cockpit is illuminated with blue-green light, which is complementary to the red-sensitive goggles, enabling him to peer beneath the goggle eyepieces to look into the cockpit and read the instruments.

These conditions are provided in the simulator by enclosing the cockpit with a light-tight box, and arranging correctly filtered cockpit lighting so that the pilot can use active night-vision goggles. The external scene video signals are manipulated to create a monochromatic sensor HUD image, occupying 20 by 20 degrees field of view, which is bordered by a very low luminance red scene. The latter is too dim to be seen by the naked eye, but it becomes appropriately bright when intensified by the goggles.

The cockpit is fixed, and the feel of the controls is simulated by a hydraulic control-loading system. The layout is based loosely on the Harrier GR5, but it is much simplified and contains a central collimated head-down display. The aircraft performance and control characteristics also resemble a GR5 in wing-borne flight. Infrared cameras enable the pilot's movements to be monitored and recorded unobtrusively.

The external scenic image is obtained from a colour television camera which is driven over a brightly illuminated 2000:1 scale model by a gantry mechanism. The effective flying area is extended by controlling the camera motion so that it is reflected at the model edges, as though the aircraft continues into the contiguous terrain seen in the vertical mirrors surrounding the model. At present a 25-fold increase provides flight over a 95 by 36km area. During the short transitions at the model edges the inappropriate camera image is suppressed, leaving the pilot with the visual sensation of suddenly encountering dense cloud. It is also possible to control the degradation of the camera image to simulate obscuration effects of low cloud and fog. Contact between the delicate camera probe and the terrain model is prevented by a micro-processor which accesses a height data-base and applies a downward limit to the vertical camera drive servomechanism.

A simulated carbon dioxide laser terrain avoidance system is included. This acts like a range-finding laser which can penetrate cloud or fog, to examine the vertical profile of the hills ahead of the aircraft. It then computes the instantaneous change in climb or dive angle to achieve a safe ground-hugging path, and this is displayed as advisory information on the HUD.

To induce the pilot to fly tactically a dense array of radar-directed missile and gun defences can be simulated. The pilot sees the missile tracks and the gun fire, whilst the head-down display shows him the tactical situation as interpreted by the aircraft radar warning receiver. He can employ a range of countermeasures, in addition to flying low enough to cause the ground radar to break lock through terrain obscuration or cluttering effects.

Arrays of tank targets have been attached to the model terrain. These miniature models, made from transparent acrylic so that they are normally unnoticeable, are illuminated by individual optical fibres so that their conspicuity can be controlled automatically. The number and identity of those illuminated is chosen to produce the required formation, for instance as a column or an assault group, and to alter the alignment of the formation relative to the aircraft approach direction. A variety of air-to-ground target designation and weapon fire control systems are also simulated.

Fig 1 shows the main functions of the units which make up the facility. There are five interconnected mini-computers with attached floating-point processors, together with the cockpit, the control-room and the terrain model assembly. Gamma computer, which models the behaviour of the aircraft, can be considered the hub. The control room contains a large equipment rack for the computer terminals and the video manipulation electronics. It also acts as an experimental control station.

3. General trials procedure

Most of our assessments follow an experimental paradigm which is aimed at obtaining statistically interpretable performance measurements. For this it is necessary to formulate detailed objectives, and decide upon the major factors which will be manipulated (the experimental variables), and those which specify the background conditions (the experimental constants). The detailed tasks, measures of performance and balanced experimental design then follow. In practice we prefer to embed the task into an assumed mission, and avoid having the pilot fly the same route repeatedly by devising as many routes as combinations of the variables. Because there are no strict repetitions, it is necessary to treat the variation between pilots as a baseline random factor in the subsequent analysis of variance.

The simulation is prepared and shaken-down by the engineering staff and test pilots, to produce a task which is credible to the trial pilots from operational squadrons. At this time it is necessary to devise a training schedule, write the briefing notes and produce an introductory video presentation which sets out the background and details of the trial. The de-briefing questionnaire is also written.

The experiment is then run with the participation of between 10 and 20 experienced ground attack pilots. Each pilot usually attends for two days during which he becomes familiar with the simulator, practises the specific trial tasks, conducts the assessed runs, and completes a detailed de-brief. When finished, the performance data are collated and subjected to statistical analyses, and the pilots' opinions about the simulation and the main experimental factors are also collated. The conclusions are extracted and then reported.

4. Examples of completed exercises

When the carbon dioxide laser terrain avoidance system had been introduced into the simulation, a trial[1] was conducted to enable pilots to compare two alternative methods of presenting the climb/dive advice, or command, using the head-up display. One technique used a pair of horizontal bars on either side of the aircraft velocity vector symbol to show a safe height corridor, and the other had an arrow pointing vertically up or down to indicate the severity and direction of the command.

The background conditions were set up to provide a low level navigation exercise at night, using a thermal image on the HUD as the primary source of visual information. During the preparations it became apparent that although the pilots were likely to have a preference for one of the display techniques, it was very unlikely to affect their height-keeping performance. However, it was thought that the usefulness of the climb/dive advice would depend strongly on the pilot's ability to interpret the shape of the ground from the thermal imager. Visibility was controlled to give a forward view of either about 1.5 km, or about 6 km. The manipulated variables then became the presence or absence of the arrow command and the two visibility conditions, but both display techniques were available to the trial pilots during practice so that they would have the experience to form preferences. The trial was conducted using a balanced design involving 16 pilots.

Not surprisingly, there were several statistically reliable performance effects. Pilots flew much lower in poor visibility when the aid was available, particularly when turning. There was less benefit in good visibility, and most pilots preferred the arrow symbol, which was consistent and easy to interpret, whereas the parallel bars could be confused with the pitch reference symbols in the HUD. Almost all of the pilots suggested that flying in the poor visibility condition was not operationally tenable, and that their technique placed great confidence in the laser aid which, in reality, would need proven integrity to justify such reliance.

A subsequent exercise was conducted to demonstrate the laser terrain avoidance system and assess whether it could compensate for the difficulties of flying through defended air-space at low level at night, when pilots would wear night-vision goggles(2). The appropriate circumstances were set up, but rather than assess the pilots' height-keeping ability, their exposure to tracking radars was considered to be more operationally relevant.

All of the pilots considered that a terrain indicating aid provided a valuable reinforcement to their perception of the terrain shape. This became essential in poor visibility, as shown by a significantly reduced exposure to the ground threats when flying with the laser aid. However, although there was a general consensus that the simulation provided an adequate opinion-forming experience, it was evident that some pilots did not react to the simulated hazards with the caution they were intended to invoke. Some also considered that they were not given the appropriate visual sensations of very low level flight, and that this affected their ability to carry out the task demands.

Other trials have been conducted to assess display devices. For instance, the benefit of a wide angle HUD for night flying is that it can present a larger thermal image, and alleviate the sensation of staring at the world through a narrow porthole. However, obtaining objective measurements which allow the benefits to be quantified is not so easy, and a trial was set up with this objective(3). Again, a balanced experimental design was used, and the only variable was the field of view of the HUD, which either represented the narrow field of current devices, or a substantial increase representing the likely limit attainable with diffractive optics.

In this case there was overwhelming subjective opinion from the pilots that the wider field of view was beneficial, enabling them to fly at the requested low level, manoeuvre the aircraft, and recognise features on the ground with much greater ease and confidence. There was also the suggestion of an objective improvement in their height, track, and speed keeping abilities with the increased field of view. However, none were statistically reliable.

5. Problems and limitations

It is tempting to assume that, having endured the processes of devising, preparing, conducting and analysing an experiment, any conclusions would be a definitive answer to the topic addressed. However, it would be an unusually gullible user of the results who would not be sceptical. In general most readers want to know whether the conclusions are likely to hold in real flying conditions. Unfortunately, there is no simple answer to such an uncertainty.

5.1 Physical inautheticities

Part of the uncertainty about the reliability of any conclusions may be due to the difference between the simulated physical conditions and those which occur in real flight. This is a huge topic (see for instance Ref 4), and the time-honoured approach has been to minimise the crucial differences using the best current technology. For instance, if the study compared the ability of pilots to use different control devices like a helmet-mounted sight and a hand-controller, then reproducing the relevant movement cues and disturbances in important. Similarly, comparison between alternative display devices needs the appropriate visual conditions such as sunlight, motion disturbance and focal accommodation.

The chief difficulties here are judging which physical factors are important, and anticipating the sensitivity of different pilots to the simulation errors. Hence it is reasonable for any conclusions to acknowledge any significant departures from real conditions, to state the opinions of the subject pilots, and to gauge the impact of the deficiencies on the results.

5.2 Psychological uncertainties

Besides such physical disparities there are a range of aspects which best fall under the umbrella of 'psychological' factors and which are likely to impose the largest uncertainties on the usefulness of any results. Paradoxically, the main incentive for our wanting to conduct simulation experiments, that of creating manipulable, operationally relevant flight hazards and tactical complexity, also brings an intrinsic psychological disparity. Although the simulation attempts to supply the salient cues and complexities of such circumstances, it cannot evoke a genuine perception of vulnerability.

For low level missions carried out by high speed aircraft, contact with the ground has a P_k of unity, whilst interaction with defences, although not so reliably lethal, is also very likely to curtail the future. A pilot's chief concern is to fly low and fast enough to evade threats, but not so low that the chance of crashing becomes unacceptable. He balances the risks dynamically, taking into account on one hand the types of defensive system, their density, placement and assumed effectiveness, and on the other the nature of the terrain and vertical obstacles, moderated by his uncertain knowledge of both.

In our experience, some pilots crash the simulated aircraft at a frequency which would decimate any airforce in weeks. Some crashes undoubtedly stem from physical limitations, such as the suddenly encountered cloud used to mask the transitions between the mirrored worlds in our flying area, or the unfamiliar handling characteristics of the modelled aircraft. Although others come from the preoccupation with incidentals, which is to be expected during the initial practice sessions for an experiment, the majority happen during assessed runs, and are most likely to arise because those pilots have an inappropriate regard for the consequences of ground contact.

This was exemplified in one experiment on sensor-assisted night flying where it was necessary to conclude that, certainly in the absence of the simulated aid, the flying technique adopted by some participating pilots was not operationally feasible. It relied on very rapid reactions to unreliable visual cues and an unlikely knowledge of terrain shape. This resulted partly from inexact briefing of the pilots, and partly because we set the minimum clearance between the delicate camera probe and the model too high, which reduced the visual sensations of imminent impact. For some pilots the mismatch, between what they were asked to do and the quality of the simulation, overstretched the simulator credibility and they ceased to regard the exercise as valid. As discussed below, the avoidance of such inappropriate behaviour may provide the key to obtaining most value from exploratory simulation experiments.

Other human factors experimenters have argued that any study conducted in the laboratory has very limited validity in predicting the usefulness of equipment or techniques in the stresses of a genuine conflict(5). They assert that it is necessary to induce the irremovable stress, fear, apprehension or general sense of vulnerability in subjects, and that any technique which effectively induces a stress reaction can be applied. They suggest, for instance, applying small voltages to electrodes on the subject's teeth, or invoking emotional reaction by post-hypnotic suggestion, or using systematic verbal abuse. Applied in the context of our studies, the stress could be advocated to generate both a general state of arousal representing combat readiness, and, if applied after ground contact or anti-aircraft weapon damage, a specific disincentive.

I am reluctant to follow such arguments on two grounds, the first is methodological, and the second is practical. To comply with the former, it would be essential to know how much stress one was applying to the particular participating pilot, and make this in some way appropriate to the assumed genuine combat level. It would be necessary to investigate a number of stress inducement techniques, assess how they affect different pilots, and calibrate their sensitivity to the principal control variables. For instance, a graph relating some measure of stress or fear reaction with the voltage applied to the teeth electrodes, for a representative group of pilots, would be a minimum for such a technique. However, it would also be necessary to know how this varies with time of day and repeated application, and whether anticipation is worse than the event. This is a lot of work.

I also have the disquieting feeling that stress is not a simple monotonic phenomenon but a complex multi-faceted mixture of physiological, cognitive and emotional elements. It therefore matters how it is evoked. For instance, a pilot who is genuinely stressed by the fear of an electrical shock would be mentally diverted into minimising his reaction, and into considering the experimental validity of the technique. Although no-one would doubt the relevance of combat stress as a factor influencing the usefulness of cockpit equipment, I feel that inappropriate artificial induction techniques could perturb the simulation rather than make it more realistic. In this sense it is like the simulation of movement, which is also a powerful influencing factor, but the absence of motion is preferable to the provision of limited body accelerations which conflict with visual cues.

From a practical viewpoint, assuming that a technique had been developed and calibrated, and that there was sufficient data from genuine battle conditions to enable choice of suitable stress levels, the prospect of enduring artificially induced fear would exacerbate the difficulties of recruiting pilots for trials. It may also destroy their acceptance of the simulator because they would, not unnaturally, be reluctant to receive unwarranted 'punishment'. Most pilots are already sufficiently sensitive to what they perceive as simulation faults, and they are usually unequivocal in saying so.

5.3 Methodological problems

Although the test pilots and squadron pilots who take part in our trials are highly professional and well motivated, the actual criteria they apply during assessed runs can have a gross effect on the outcome. Such criteria are not overt, and there may be a considerable difference between what any individual will declare, acknowledge, and apply.

One uncertainty arises from our preferred design of experiment, whereby all subjects experience all of the trial conditions so that they can formulate informed opinions. Knowing that he is going to be assessed over an extended period, of a few days intensive flying, it would be quite reasonable for a pilot to develop a strategy for accomplishing the task requirements which he could sustain in any of the experimental conditions. He could, tacitly, pick a level of performance which he judged the experimenters would deem satisfactory, and differences between conditions would be evident solely to the subject as requiring less effort, rather than as manifest changes of his performance. The primary results are then the subjective assessments of the relative ease or difficulty of the alternative conditions.

In anticipation of this methodological effect, and knowing that the inter-pilot differences will always tend to swamp the subtler experimental factors in any statistical analyses, it is very tempting to set up exaggerated conditions. For instance, by making forward visibility either extremely poor or optimistically good, it may be possible to force demonstrably significant objective results. However, the conditions could easily cease to represent operational concerns, and the pilots may reject the inverted circumstances as credible.

Other facets could tend to annul significant performance effects. Increasing the complexity of the simulation, so that it better resembles the operational concerns of the pilot during real missions, opens opportunities for individuals to trade-off their perception of the component task priorities against their specific abilities. Thus a pilot who can fly very accurately at low level may prefer to devote his attention to achieving a good height-keeping score, to the detriment of maintaining track or managing the weapons. Another pilot may apply different priorities. Such inter-pilot variation is intrinsic to real complex operations, and the simulation study must accommodate the individual pilot's freedom to choose a strategy which best suits him. The net effect, taken over all the participants, is to spread the variation of the component task scores, so that each is less likely to show effects attributable to the main variables of the experiment.

Finally, it is not unknown for subjects to make use of tricks which are only possible in the simulator to obtain easier or better performance. A classic illustration of this occurs in air-combat simulators where the target aircraft image is projected onto the inner surface of the surrounding spherical screen. When the pilot loses sight of the enemy aircraft the effort and delay of carrying out a systematic search can be circumvented by looking up to see where the projector is pointing. Inappropriate behavioural strategies for exploiting knowledge of the engineering techniques and limitations can be used in all research simulators, but most would be less obvious than this.

6. Extracting useful conclusions

In the light of the methodological problems of arranging suitable experiments, and acknowledging the intrinsic physical and psychological limitations, how can simulation yield useful results? I think the key issue is the behaviour of the trial pilots since, in the foregoing discussion, the facet which consistently indicates anomalies between simulated and real circumstances has involved the pilot acting inappropriately.

That pilots accept the simulation and engage themselves *actively and positively* in the hypothetical circumstances created, is of fundamental importance to the conduct and usefulness of the experiments. Any pilot knows that there is no real risk, and that crashing or being shot down in the simulator are personally inconsequential. However, the familiarisation, training, briefing and practice periods should be arranged to encourage him to immerse himself in the simulated mission context. The best he can do is attempt to act as he hopes he would in a real mission, accepting the tasks and the physical representation of the aircraft, its systems, the visual conditions, hazards and threats as a hypothetical dynamic interactive circumstance.

This is much like the reader of a novel who can accept invented characters, context and events with an attitude of selectively suspended disbelief, while still applying common sense and experience to judge the credibility of behaviour. Simulation and fiction are similar in that they are good only if they stimulate the wish to enter the created world, and they must then sustain their hold. A self-honest trial pilot in a research simulator will bring his operating experience and skills, and attempt to apply them to the successful accomplishment of the specific trial tasks. How well he does, and what opinions he has, are as important as ever, but understanding how he does so is also very valuable.

The main assertion is that if a pilot devises a strategy for accomplishing the trial requirements which shows an effective incorporation of the novel, hypothetical elements into his accumulated airmanship, flying, navigating, target finding, weapon aiming, threat avoiding, and mission managing skills, then this strategy is likely to apply in real flying. Consequently, differences in his ability to use alternative techniques, or equipments, or to operate in different conditions, are only likely to apply if his behaviour meets this criterion. It also follows that exploratory simulation studies should allow the participating pilots to develop strategies for using the novel equipment, procedures or tactics.

Understanding the techniques which pilots employ is one of the primary outcomes of the experiment, and it is especially valuable when attempting to extrapolate conclusions to real flying. The initial familiarisation and briefing should induce pilots to regard crashing or engagement by threats as consequential, and exploiting anomalies of the simulation should be anticipated and discouraged. Then, for each pilot it is necessary to decide whether:

(a) his technique exploits simulation deficiencies, or he has neglected the briefed criteria,

or (b) he has attempted to apply realistic criteria but the simulator has not enabled him to do so,

or (c) he has used realistic criteria and the simulator has not limited his technique significantly.

Adopting this approach does not alter the need to design balanced, statistically interpretable, rigorous experiments. If, however, a pilot completes the assessed runs and his behaviour falls into category (a) then measurements of his performance are not relevant. An additional pilot should repeat the same sequence of experimental conditions in order to generate satisfactory data. As noted in 5.1 above, behaviour conforming to category (b) is, in a sense, a failure of the simulation to provide the relevant conditions, and this should be made explicit in any experimental conclusions. Only behaviour conforming to category (c) is likely to hold in flight. The usefulness of the experimental results can thus be judged by the proportion of each category.

We now have the problem of attempting to understand an individual pilot's operating strategy so that it can be assigned to one of the above classes. Thus the debrief which follows the assessed runs should at least ask pilots about their operating techniques and the influence of the salient simulation factors. It is also very valuable to obtain video recordings of overt behaviour, such as head, hands and eye movements, which enable operating strategies to be inferred. Recording eye direction of fixation is a particularly powerful technique, because it shows with high temporal resolution how the pilot attends to the competing demands of all the mission-related concerns. Quantitative analysis may also enable his priorities to be inferred. Unfortunately, such recordings are difficult to obtain and time-consuming to analyse, but they can be made using a few representative pilots outside the main assessment. It is, however, of considerable assistance to the whole process for all pilots to give a running commentary describing what they are doing as they do it.

7. Conclusions

(a) RAE Farnborough has developed a facility for studying the operation of novel cockpit, avionics and weapon systems in day or night, variable visibility conditions, during low level high speed flight across hilly terrain containing vertical obstacles and radar-directed threats.

(b) Balanced experiments have been conducted on a range of topics using experienced ground-attack pilots to derive comparative performance measurements and subjective appraisals.

(c) The results are only likely to be capable of extrapolation to real flight if the participating pilots consider that the physical limitations of the simulator do not affect their working technique significantly, and they are as cautious and concerned for safety as they would be in genuine flight.

(d) It is of great value to observe, understand, and report the behavioural strategies developed by trial pilots.

References.

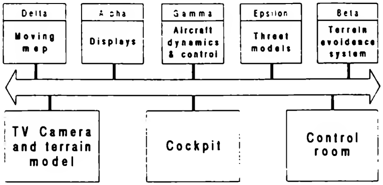
1. J.I. Askew, N.J. Benger, D.N. Jarrett and P.R. Laughton, 'Simulator assessment of a laser-derived director to aid terrain following during night ground attack missions', RAE Technical Report (in publication).

2. N.J. Benger, D.N. Jarrett and Sq Ldr D. Jackson, 'Assessment of a terrain following laser: summary and discussion of simulation trial NATAL', RAE Technical Memorandum (in publication).

3. G. Jones, A.R. Buffett and D.N. Jarrett, 'Assessment of a wide field of view head up display during night ground attack missions: preparations for a simulator trial', RAE Technical Memorandum FS(F) 507 (1983).

4. J.M. Rolfe and K.J. Staples (Eds), 'Flight Simulation', Cambridge Aerospace Series, CUP, Cambridge (1986).

5. P. Watson (quoted in) 'War on the mind: the military uses and abuses of psychology', Hutchinson, London (1978).



The Ground-Attack simulation facility

AMERICAN INSTITUTE OF AERONAUTICS AND ASTRONAUTICS
370 L'Enfant Promenade, SW
Washington, DC 20024

88-4604-CP

TACTICAL COMBAT SIMULATION ENVIRONMENT

Barry W. Webb*
Lockheed Aeronautical Systems Company
Marietta, Georgia

John W. Peek **
UNISYS
Marietta, Georgia

Abstract

As new technologies emerge and mature, it is important to explore their application within the aerospace industry. Lockheed Aeronautical Systems Company (LASC) is currently exploring the use of Artificial Intelligence (AI) for improved mission effectiveness and enhanced survivability of fighter aircraft. To support the research, integration, and testing of such high technology programs, LASC developed its Tactical Combat Simulation Environment (TCSE).

This paper gives an overview of the LASC TCSE, including its history, current configuration, and future enhancements. It also covers the context for the development of the TCSE, the varying levels of "real-time" simulation available, and the features of the major simulation components: ownship aerodynamics, threat environment, sensor data management, stores management, voice input/output, display generation, global battle display, and intermessage transfer, recording and playback.

Introduction

The application of AI for mission enhancement addresses the impact on mission effectiveness and survivability (within manned vehicles) resulting from an excessive quantity of information in the modern air combat environment. During a mission, the rapidly changing threat situation produces an enormous amount of information for the avionics systems within modern aircraft. These systems often provide excessive or irrelevant information at the most critical moments, overwhelming the pilot. This reduces pilot performance due to loss of situational awareness, thus reducing the overall mission effectiveness and survivability.

Avionics systems incorporating AI technology are needed to supply pilots with carefully filtered and fused data concerning situational assessment, prioritized feasible responses to threats, and other automated functions. Such systems cannot be tested in isolation, nor can they be fully evaluated with limited functionality test cases. To determine their potential contribution to mission effectiveness and survivability, airborne expert systems must be evaluated within a rich, high-fidelity simulation environment tailored to handle embedded AI-based systems. This environment must be capable of supplying the expert system with all the stimuli (i.e. autopilot, flight management computer, fuel, sensors, weapons, etc.) that would be available in the "real" aircraft. The environment also must be able to heuristically reconfigure displays based on display resources, mission phase, and pilot preference. It must be able to simulate a rapidly changing combat environment, including both the environment external to the aircraft and the environment within the aircraft.

Background

Conceived in 1986, the TCSE began as a simple, economical environment for testing and evaluating the enabling AI technology. The foundation was laid to create a generic, reconfigurable cockpit, cockpit display system, intermessage transfer system, and a set of generic aircraft aero models. Figure A contains a diagram of the 1986 TCSE configuration. Subsequently, more sophisticated features were added including simulation of ground and airborne threats, sensors, ownship faults and status changes, stores management, global battle display, and threat station consoles. These additions created a test environment rich in capabilities, yet flexible and reconfigurable. Figure B contains a diagram detailing the 1988 TCSE configuration. The remainder of this paper describes the current TCSE configuration and the enhancements planned to carry the TCSE into the 1990s.

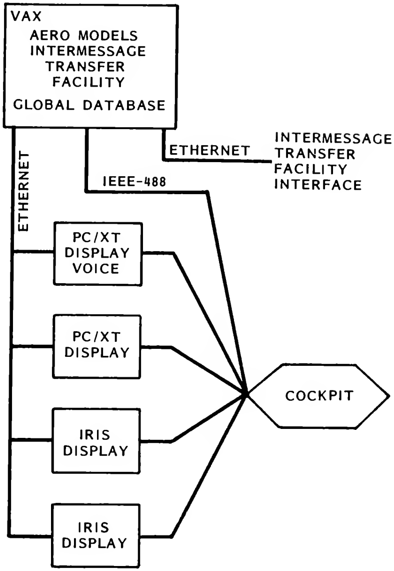


Figure A. 1986 TCSE Configuration

Intermessage Transfer Facility

A large simulation environment that is expected to communicate with numerous and different expert systems must have a configurable and robust communication capability. The Intermessage Transfer Facility (ITF) provides such a capability. It was anticipated that support for

[*] Scientist-Associate [**] Principal Scientific Analyst

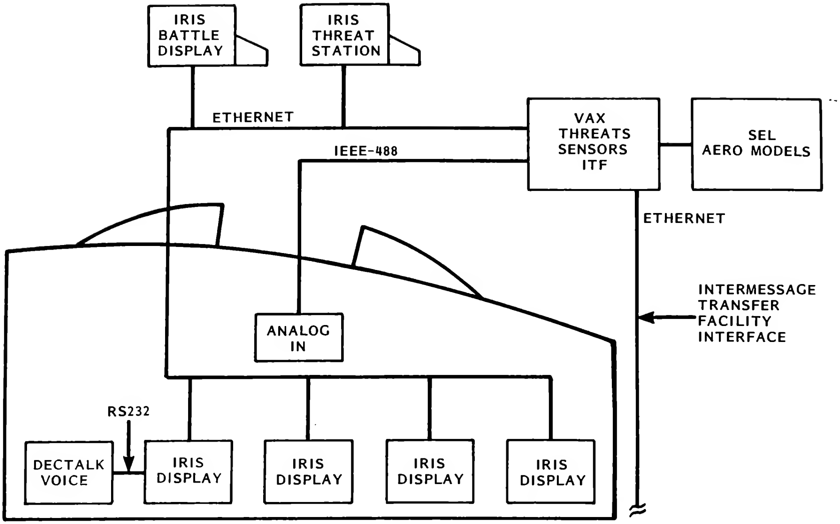


Figure B. 1988 TCSE Configuration

Ethernet and its various protocols would be the best communication medium since most computers used for research have an Ethernet interface. In this light, the ITF was designed to support Ethernet with the major communication protocols used today: TCP/IP, XNS, DECnet, Chaosnet.

The first role of the ITF is to provide a bridge between different networks with varying protocols. Messages may be sent between any two or more computers without regard to protocol. Computers with like protocols may interchange messages directly without any intervention from the ITF.

The second role is to create and send messages that contain specific information on request and on a synchronous basis. Simulation state values and control messages are generated by the ITF and sent to the requesting computer. The format and content of these messages are predefined before runtime and can be easily modified. In addition, data and control messages may be generated and sent to the ITF from any computer to provide information to the simulation environment. Since many applications cannot maintain consistent "real-time" speed (first level of "real-time" execution), a start-stop mechanism (second level of "real-time" execution) is provided through a control message named SIM-FREEZE. When an application desires the TCSE to suspend execution, this message may be sent via the ITF. Once ready to continue, the application may generate and send a SIM-RUN message to restart the simulation. ITF-generated messages destined for multiple computers are broadcast to eliminate protocol overhead. Because there is no guaranteed receipt of these messages, they are closely scrutinized before runtime to ensure their loss will not cause major problems. Synchronous messages containing non-discrete information are the most likely candidates for broadcasting.

The third and last role of the ITF pertains to message insertion, recording and playback. During tests when all computers are not running, the ITF allows a researcher to enter a test message that will be sent to the appropriate destination(s). With more than 20 computers at hand, such a facility eliminates the need for everyone to be involved and greatly increases development productivity. In addition to message insertion, message recording can be invoked. All or selected messages can be recorded during runtime for later analysis. Since most messages are ASCII in nature, they can be easily read by the researchers. Message playback (third level of "real-time" execution) from the ITF has greatly enhanced the TCSE's demonstration capabilities by allowing previously recorded messages to be retransmitted. In this mode, the ITF's playback (of a test run that used the start-stop mechanism) allows researchers to view the data in real-time. This feature above all has proven to be indispensable.

Aero Simulation

Ownship

The aero/flight controls/propulsion simulation is a 6 degree-of-freedom point-mass model of a generic fighter aircraft. The models contain all routines and data necessary to simulate the aircraft in basic flight. The equations of motion, atmospheric, and inertia models are included. The simulation trims the aircraft as part of the initialization process and freezes with the aircraft in straight, level flight. It is organized using data arrays with an array processor emulator to reduce execution time.

The simulation has a few significant omissions. The flight controls model is limited in that the control stick input determines the aircraft acceleration and the throttle input determines thrust and fuel flow. The means of obtaining the responses (control surfaces and engines) are not specifically modeled. The modeling of aircraft weight during flight is accomplished through the burning of fuel and releasing of stores.

Autopilot

The autopilot model is set to be on at all times. The default mode of operation is Pitch Attitude Hold for the vertical axis and Roll Attitude/Heading Hold for the horizontal axis. This is the normal Control Stick Steering (CSS) autopilot option. Changes in desired attitude are commanded through the stick as in normal flight. The autopilot will hold the aircraft attitude at stick release; and with the wings banked less than 2 degrees, the autopilot will hold the aircraft heading. Upon engaging the Altitude Hold mode, stick inputs in the vertical axis will be ignored. The reference altitude is the pressure altitude at the time of autopilot engagement. Engaging the Autonav mode causes both the pitch and roll commands to be generated based on error signals received from the Flight Management Computer (FMC). For this mode to work properly the flight plan must already be entered into the system, and the selected navigation mode must be in the FMC. While flying in the Autonav mode, all stick inputs are ignored. The mode is exited by selecting either Altitude Hold or Heading Hold. Here, selecting Heading Hold will cause reversion to the full attitude hold in both axes while selecting Altitude Hold will cause altitude hold in pitch and attitude hold in roll.

Flight Management Computer

The Flight Management Computer (FMC) model continuously determines the aircraft's position based on data from the three inertial and airdata systems, supplemented with positioning data from the VOR/DME (VHF Omni Range/Distance Measuring Equipment), MLS (Microwave Landing System), GPS (Global Positioning System), and manual inputs of ground reference points. The guidance parameters are derived from the entered flight plan and the difference between the actual and desired groundtrack, vertical profile, and time en route. The guidance commands are provided to the flight director, autopilot, and auto-throttle by the appropriate selection.

Fuel Management System

The fuel system is designed to work automatically through a Fuel Management System (FMS). In the event the FMS is lost, normal fuel system operation can continue through pilot control of appropriate fuel pumps. The FMS transfers fuel from appropriate tanks to the main fuel tank to maintain an optimum center of gravity for the best maneuverability. During air refueling, the FMS allows fuel to be onloaded to the appropriate tanks to maintain a center of gravity that permits best stability/maneuverability for that flight regime. Tanks are scavenged to minimize unusable fuel. Each tank has a level sensor, boost pump, flowmeter, pump status indicator, and shut-off valves. All fuel lines are redundant, separated sufficiently to prevent loss of both lines to a given tank due to hostile-induced damage.

Sensor Simulation

The simulated sensor suite of the TCSE is adequate for most applications. Most of the sensor models use a data file (track file) to make updates on each object they are tracking. As changes occur, the sensor model updates the track file with the current information (e.g., time-last-seen, position, velocity, altitude, etc.). These files are used as input to other models which fuse the data into the best composite picture that can be derived from multiple sensors and platforms. The following paragraphs give a short synopsis of the TCSE's sensor and fusion models whose logical flow is depicted in Figure C.

Sensor Data Manager

During the simulation's initialization phase, an operator-controlled scenario file is scanned for track input. This file allows the operator to establish the complete scenario and beginning mission time from which the Sensor Data Manager (SDM) will calculate relative track positions. The scenario file contains the definition of all air and ground objects used during the planned mission. Examples of track attributes include mission start time, radar mode, waypoints to follow, velocity and altitude to attain, and weapon stores if appropriate. If the simulation needs to be started at some point after the beginning mission time, the SDM calculates the elapsed track positions before initialization is complete. The SDM answers information requests from each individual avionic sensor; responds to modes, commands, and cues; and displays information in the cockpit. The SDM aids the simulation effort by performing search definition and sensor track management. Search area definition involves management of the search control parameters for the steerable sensors. The pilot sets the desired search mode, operating volume, and frame times. The SDM prioritizes tracks for allocation and cuing to individual sensors. Tracks are stored by identification, range and time-first-seen. Once the sensor information has been prioritized and assigned to tracks, sensor requests are sent out for implementation. The SDM transmits search volume, location, and frame times together with the prioritized, allocated implementation requests. The following paragraphs describe each of the sensor models encompassed within the SDM.

Threat Warning System

The Threat Warning System possesses capabilities for threat detection, analysis, identification, and countermeasure determination using data such as threat location, type, power, and mode. It can simulate the detection of all known tracking radars used for fire control of air-to-air weapons and surface-to-air missiles. The Threat Warning System also includes laser ranging and fire control signals, and provides laser warning receiver simulation and associated electro-optical sensors.

Radar Warning System

The Radar Warning System detects active bodies such as friendly or hostile aircraft, missiles, and possible threats. In addition to detection, the model measures azimuth, elevation, and track radar sources. The model obtains necessary information from the ground control intercept, early warning radars, missile launching sites, missiles in flight, and aircraft including the Advanced Warning and Control System (AWACS). The Radar Warning System forms a track file for use in assessing threats and discriminating friendly sources. The model accounts for the effects of received signal strength, terrain masking, emitter characteristics, and field-of-view for both the transmitter and receiver. The track file accounts for apriori data, range, threat type and mode.

Ultra-Reliable Radar

The Ultra-Reliable Radar (URR) simulates the operation of both monostatic and passive bistatic radar systems that are dependent on illumination of the target by noncooperative ground-based radars. The model detects and tracks targets within the desired defined search/track volume. Range, bearing, altitude, raid assessment, and possible identification are all attributes which this model can derive. Target position for the bistatic radar is calculated by triangulation between the illuminator, the receiver and the target.

Missile Approach Radar

The Missile Approach Radar receives cues whenever a missile has been launched or detected. It detects missile launches through the observation of missile range and rate. The radar system creates an entry in its track file for each individual missile, incorporating estimated position, velocity, and acceleration once the missile has been detected. When the radar performs subsequent detections, the model updates the corresponding track file entries.

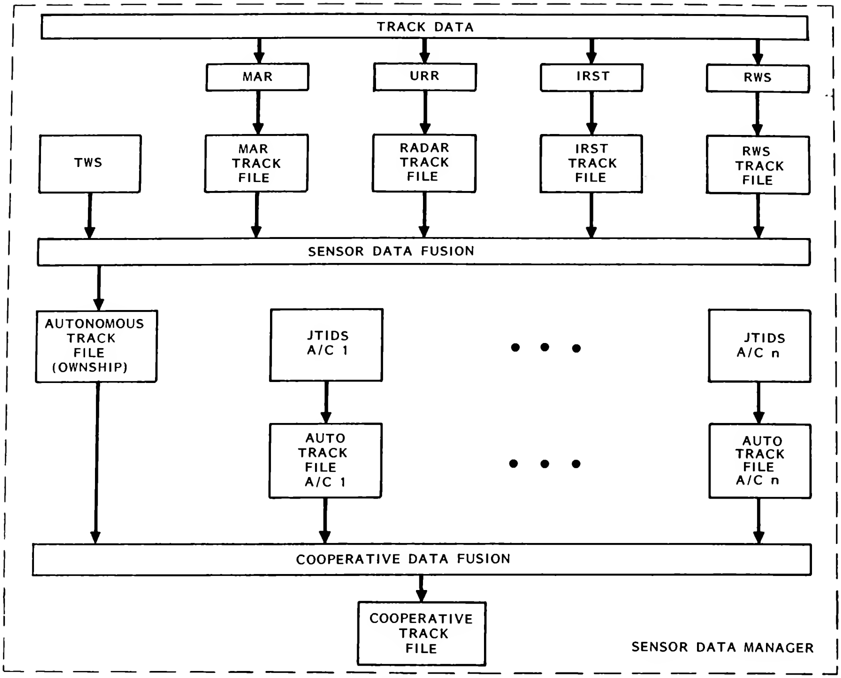


Figure C. Sensor Fusion Logic Flow

Infrared Search and Track

The Infrared Search and Track (IRST) model provides passive long-range detection, raid assessment, passive ranging, and high-accuracy angular target tracking. In all search modes, the coarse for up to 10 targets can be maintained. The IRST is stablized in several ways: (a) along the longitudinal axis of the aircraft (ownship referenced), (b) in roll and pitch (horizon referenced), (c) in roll, pitch and yaw (Inertial Navigation System referenced), and (d) in its plane as derived by the line-of-sight and the relative velocity. Once the threat has been detected, the IRST system creates an entry in its track file for each individual threat, incorporating estimated position, velocity, and acceleration. When the IRST performs subsequent detections, the model updates the corresponding track file entries.

Sensor Data Fusion

The Sensor Data Fusion model provides the best possible estimate of the threat environment including identification and kinematics. The model combines all the information received from the various onboard sensors and incorporates the information into a data file (Autonomous Track File) accessible by the pilot and the ITF. This method ensures the proper situational awareness, (e.g., in a beyond-visual-range (BVR) phase of attack), as well as the most efficient use of sensor information. The model continually updates the Autonomous Track File to provide the pilot with the most accurate data for situation assessment.

Communication Systems

The Communication Systems model permits information exchange between the pilot and the outside world by both basic radio communications and a simulated Joint Tactical Information Distribution System (JTIDS) data link. The UHF/VHF radio system accepts pilot input for band, frequency, mode, radio selection, and failure status. The mode, frequency, and failure status appear on the pilot's displays. The JTIDS data link simulates the time-based exchange of tactical information between the simulated ownship and numerous external platforms. JTIDS provides tactical information to the pilot displays and ITF, taking into account tactical location, identification, and target priority information supplied by the simulation. JTIDS also allows the pilot to update existing mission databases. This Information is useful to the pilot for coordination and confirmation of existing targets and possible future weapon choice and firing. JTIDS determines which information can be received by ownship based on the relative position of all other JTIDS-transmitting aircraft.

Cooperative Data Fusion

The Cooperative Data Fusion process is modeled by taking an inclusive "OR" of the global air and ground objects that are "seen" by the Individual cooperating JTIDS platforms. For each global air and ground object attribute, the platform whose Autonomous Track File supplies the most accuracy and resolution for that attribute is selected to provide the value. If two platforms are tracking the same object, then the range value may be supplied even if the individual platforms have angle information only (i.e. it is assumed that triangulation is possible). The Autonomous Track File, created as a result of the Sensor Data Fusion process, is then fused with the JTIDS results to produce the Cooperative Track File. This is the final track file produced from which the pilot and ITF may access data with the highest fidelity.

Weapon Systems

The TCSE provides all the functions necessary to select, arm, and employ various ownship air-to-air and air-to-surface weapons. It uses two models that monitor weapon selection and functioning: the Stores Management model and the Fire Control model. The Stores Management model determines weapon status, identification, type, arming, and fusing status based on selections made by the pilot or through the ITF for each individual station. The Fire Control model determines the maximum and minimum range, steering, and time to fire for delivery based on the particular weapon type, quantity, and employment modes selected. This model performs calculations for air-to-air missiles, air-to-surface missiles, and guns firing at selected targets based on weapon mode, type, and target selected. Weapon delivery calculations include gun solutions as well as lead angle. The following paragraphs describe each of the weapon systems modeled within the TCSE.

Air-to-Air Missiles

The TCSE's Air-to-Air missile simulation includes a model of missile aerodynamics, propulsion, fusing, and detonation to account for missile launch and guidance. The model determines weapon selection, arming, steering, and range and it updates the missile's status and directs the information to the sensor models. The TCSE operator programs the missile inventory, weapons, and stores status as part of the scenario setup. During the scenario, the missiles may be selected and launched against various targets by the pilot or through the ITF. Missile flyout equations simulate missile sensor tracking and guidance against the selected target, accounting for missile and target flight characteristics and errors induced by target countermeasure effects and the atmospheric environment.

Countermeasures

The Countermeasures model uses Electronic Countermeasures (ECM) logic combined with Radar Warning Receiver (RWR) data to simulate the aircraft's ability to counter an existing threat. Based on input from the pilot or the ITF, the model activates the ECM device. The location and type of both missile and missile site determines the effectiveness of ECM jamming.

The model also updates and displays the status and availability of chaff and flares. In employing chaff, for example, the pilot controls the size and location of the corridor. The chaff function supplies chaff status information to the sensor and weapon models, enabling them to include its effects in their detection and tracking computations. The chaff function calculates the corridor position based on the location of the threat aircraft at the time of release.

Surface-to-Air Missile Site Radar

The Surface-to-Air Missile Site Radar model simulates radar sensors for search, track, semi-active, and fully active radar sites. The simulated track radar calculates the range and direction of both targets and missiles. The radar cross-section is continuously updated to simulate loss-of-track and target reacquisition by the site as the signal changes. Tracking occurs only when the target's signal-to-noise ratio allows radar lock-on. The calculations also account for atmospheric effects such as scintillation and glint. The semi-active missile model simulates the interaction of a track (illumination) radar and a semi-active missile radar. The fully-active missile radar model simulates the illumination and detection functions of a fully active missile. The model calculates the target's range, direction, and the strength of the radar signal returning from the target. The missile is guided using the radar beam transmitted by the missile and reflected from the target. The process supplies missile data to the Missile Launch model and the Missile Flyout and Guidance model to aid in their calculations. The data are also sent to the avionic sensor simulators for detection processing. The simulation also accounts for the effects of countermeasures and atmospheric conditions on a radar's tracking and detection functions.

Surface-to-Air Missile Launch System

The missile flight process begins with the decision to launch at an inbound target based on the impact point being within the missile envelope. The Missile Launch System determines the launcher's position through its azimuth and elevation. The model contains the launch enable logic based on system type delays and target position within the missile launch envelope. Once the position has been defined, the Missile Launch System enables the launch flag. The missile launch process computes all appropriate delays, then activates the launch.

Missile Flyout and Guidance System

The Missile Flyout and Guidance model performs flight calculations for the missile to track and intercept the target. The model determines the current missile position and status. The surface-to-air target tracking radar and the missile seeker determine the current tracked position of the target aircraft. During flight toward the target, the model periodically computes missile guidance commands, continually updating the missile's aerodynamics and thrust characteristics. The missile fuses when it is within the missile type's detonation range.

Display Generation

The Display Generator (DG), an important and integral component of the TCSE, provides the capability to display information in numerous forms to the pilot. Display Elements (DEs), the graphical objects used for displaying information may be selected before runtime to provide the instruments necessary to fly the aircraft. The current library of approximately 150 DEs will more than double in size in the future. Most DEs are generic in nature and may represent numerous aircraft instruments. An example would be a moving-pointer scale that may have oil pressure bound to it as its datasource. It is possible to use multiple occurrences of this DE with different datasources. On the other hand, some DEs are specific in nature and represent only one aircraft instrument which may have one or more permanent datasources bound to it. An example would be the Horizontal Situation Indicator (HSI). In addition to predefined DEs, DG is completely controllable via the ITF to manipulate DEs and their datasources at will. This occurs during runtime to provide the capability to format and reconfigure the cockpit display system.

In order to display current information to the pilot, DG must know numerous aircraft state values, which are calculated by the aircraft simulation. To provide this knowledge, a stream of aircraft state values is sent to DG, which keeps a copy of the real world values for use as datasources. This stream determines what datasources are available for assignment to the various DEs. After a datasource is bound to a DE, DG keeps the value representation updated automatically without further instruction.

DEs are individual entities and may stand alone or be grouped as components in a hierarchical fashion. This capability allows an outside source (i.e. expert system) to reason about the DEs at varying levels. Also, this grouping automatically incorporates a minimum amount of a new feature known as Value Inheritance (VI). For example, if a parent DE is turned off, all of its children DEs are also turned off. Of course, the ability to control DEs may take place individually at all times. DEs have a set of parameters whose values are used to determine their capabilities and features. Examples of parameters include text color, orientation, screen coordinate position, magnification, etc. VI makes it possible to link parameters between different DEs so that changes to one parameter will affect many other DEs.

DG appears as one logical/physical unit to the TCSE. In reality, it is four units each controlling one cockpit display. Formatting commands are issued by the controlling source and all four units react. It is not until the command to assign a DE to a specific display surface is issued that the four units react differently. This capability allows DG to display all DEs used during runtime on any display surface selected.

Two other capabilities within DG are worth noting. First, DG has a voice synthesis capability. Currently, this is performed by a DECTalk device. Messages to be spoken are sent through the ITF and are formatted and handed to the DECTalk device. Second, the display surfaces within the cockpit have touch-sensitive overlays connected to DG. Menus may be presented to the pilot, who simply touches the glass to make selections. See Figure D for a diagram of DG's functionality.

Although quite elaborate and robust, DG was created with expandability in mind. The library of DEs may be expanded or changed to support any application necessary. Since DG uses the well-defined interface of the ITF, new applications desiring to use and control DG can be integrated with speed and efficiency.

Voice Input

Currently, voice input to the TCSE is available through a stand-alone configuration using a Kurzweil speech recognition system that is serially connected to an IBM PC/XT. The PC is used to define the vocabulary templates, train the different users of the system, and provide serial output of the speech recognition results. Applications desiring speech recognition may independently use the PC's output directly through the serial interface.

Global Battle Display

Because of the need to show the "big" picture to audiences during demonstrations, the TCSE includes the Global Battle Display system. This system runs in a stand-alone mode and receives simulation data through the ITF. It provides a pictorial representation of the geographic region, political boundaries, bases and cities, all ground and air threats, friendlies, their actual identification and position, and the overall status of the TCSE. A menu-controlled system, the Global Battle Display, when combined with a large overhead television projector, provides the capability to show the entire battle area as the scenario unfolds.

Threat Stations

For the ownship to fly against predefined threats from the scenario file is one thing, but to fly against another pilot is even better. That is exactly the reason behind the threat station consoles. The threat stations were developed to allow a second pilot to fly without scenario restrictions. During the mission, they allow the pilot to select any aircraft (usually hostile) in the scenario and take over its control from the Sensor Data Manager. Once selected, the aircraft is flown from the console and has full maneuverability, radar, stores, fuel, and autopilot mode to allow the pilot to fly against the ownship. Positional data are transmitted to and from the threat station to provide a real-time update that can be seen on the cockpit displays by the ownship pilot. All air and ground objects are visible on the threat station console to provide a realistic view of the scenario. When finished, the pilot relinquishes control of the aircraft and the Sensor Data Manager resumes control.

External Interface

To ensure that the TCSE remains an "open" environment, a well-defined external interface was incorporated. This interface defines the capabilities available within the Display Generation system and the format and content of every message that can be transmitted through the ITF. Current and potential application projects are encouraged to review this interface to obtain a thorough understanding of the TCSE's capabilities. If after perusal a project finds that its needs are not met, the interface may be enhanced to provide the new capabilities. This is a never-ending process and is meticulously managed by a system administrator. Before any message is added, deleted, or changed, all side effects are researched to ensure no problems are inserted.

Tools

Global Data Structures Manager

There is, of necessity, a large amount of data that must be available to the various simulation models and to the external world in any large software system such as the TCSE. To aid in managing and maintaining the data necessary for the simulator, we have developed a utility called the Global Data Structures Manager (GDSM) and a set of preprocessors for Ratfor, C, and FORTRAN that work in concert with GDSM. GDSM automatically provides to the software designer several functions that have heretofore been manual tasks. The salient features of GDSM are that it:

- Is completely menu-driven.
- Handles multiple projects.
- Performs name collision checking for both variables and alternate names.
- Contains full maintenance capability for the global database.
- Checks length of each attribute field on input.
- Contains user restriction capability for each individual command.
- Performs automatic generation of Fortran Common Blocks and the equivalent C structure.
- Creates on-line interface diagrams.
- Contains full complement of reports for documentation.

To identify the global variable names in the original source code, all global variable names begin wlth "$." Since this is not a legal FORTRAN name, some method was needed to convert these names into an acceptable form. The pre-processors accomplish this task and include the following capabilities:

- All global variable names ($.name) are replaced wlth their alternate name.
- An include statement is inserted for the standard deflnltions and macros.
- An include statement Is inserted for each common block referenced.
- A record is updated in the module usage data file for each module that is processed.

These routines relieve the software designer from having to be concerned with identification of all the software components assoclated with a particular project.

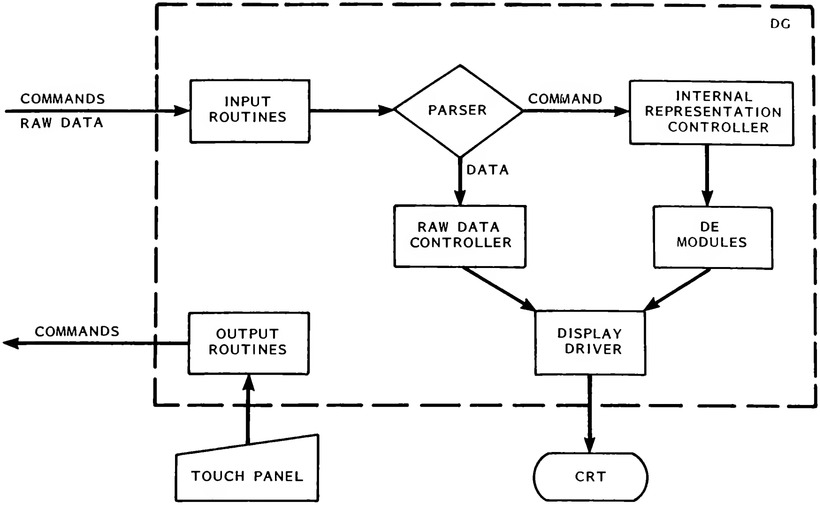


Figure D. Display Generation Logic Flow

Data Retrieval Unit

Another useful tool that works in conjunction with GDSM and the pre-processors is the Data Retrieval Unit (DRU). Named after an actual hardware device, DRU is a software package that can be used to examine and manipulate the simulation's global database during runtime. It includes:

- Examines and modifies "$." variable values.
- Handles FORTRAN or C constructs.
- Performs value representation in decimal, octal or hex.
- Performs "$." to alternate name translation.
- Performs value conversion such as radians to degrees and feet to nautical miles.

These tools as a whole have provided the TCSE with an invaluable service in the development of the simulation software. Automatic name collision checking, automatic common block maintenance, readily available reports, online interface diagrams, etc., have greatly relieved the software designers from some of the more routine tasks that are inevitably part of a large software development project.

Future Enhancements

Future enhancements will include the addition of six Gould Concept/32 9780 mainframe computers that will communicate through a Gould Multiprocessor Shared Memory System, MS2 model 3050. Four Gould mainframe computers will support the simulation of Avionics, Threats (two computers), and Air Vehicles. The Avionics computer will simulate all the cockpit avionics equipment. This equipment will include the Navigational Guidance System, the Weapons System, and the Electronic Countermeasures System. The Threat computers will simulate a hostile environment that includes aircraft, air-to-air missiles, and surface-to-air missile sites. The Air Vehicle computer will simulate the ownship's flight envelope and responses to flight control. These four computers will be augmented with array processors to support fast numerical processing. A fifth Gould mainframe will handle all I/O functions. This 9780 will act as the host computer, interfacing the major computers, the cockpit, and other smaller systems, and will also be augmented with an array processor. The sixth 9780 will be used strictly for development purposes. See Figure E for a diagram of the enhanced TCSE.

The air vehicle dynamics models will be upgraded to perform a higher-fidelity of flight characteristics. The aerodynamics model will consist of closer-to-life airframe and flight control surfaces. They will include landing gear, movable doors and hatches, aero-elastic effects, hinge moment effects, fuel transfer effects, braking chute effects, vectored thrust, ground proximity effects, and atmospheric effects including turbulence and windshear. The flight control model will further model additional sensors, control surfaces, direct force devices, inner loop control laws for stabilization, and outer loop control laws for autopilot.

A post simulation reduction function will be added to provide data reduction of pre-selected parameters in the form of printouts, plots, strip charts, and computer tapes for user analysis. This feature will allow immediate analysis of test programs by enabling decisions for the next test run based on the latest post test data received.

A higher-fidelity air-to-air and surface-to-air threat environment will be added. This environment will have a simulated centralized command and control defense consisting of air and surface defenses with coordination of enemy assests.

Conclusions

In order to test AI-based avionics systems, Lockheed Aeronautical Systems Company developed its Tactical Combat Simulation Environment. From the very beginning we believed that the TCSE must be extremely flexible to accommodate the wide range of potential client projects. In this light, every effort has been made to make the control of its configuration easy and efficient. Options to include specific computers, generate specific messages, and start the scenario where needed are all selected by the operator prior to initialization. The use of Ethernet as the medium for communications and the ITF's ability to provide a window into the TCSE have made it possible to support a wide variety of advanced technology projects. In addition, the three levels of "real-time" simulation support have eliminated the requirement for client projects to provide fast, powerful computers to interface with the TCSE. Currently, two projects have successfully used the TCSE and several others have plans for its use. The TCSE is viewed as a powerful and versatile environment for the development and testing of advanced aircraft systems.

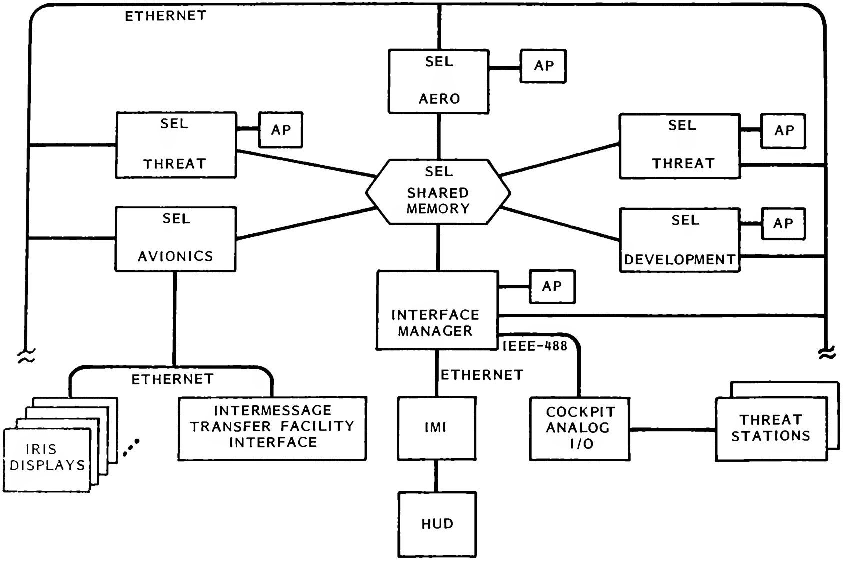


Figure E. Late 1988 TCSE Configuration

88-4605-CP

VSRA IN-FLIGHT SIMULATOR --- ITS EVALUATION AND APPLICATIONS

Masaki Komoda[*]
Tokyo Metropolitan Institute of Technology, Tokyo, Japan

Nagakatsu Kawahata[†]
Nihon University, Chiba, Japan

and

Yukichi Tsukano[¶] and Takatsugu Ono[¶]
National Aerospace Laboratory, Tokyo, Japan

Abstract

The paper describes an in-flight simulator named VSRA (Variable Stability and Response Airplane), in some detail. The VSRA system is designed based upon an explicit model following theory. Only linearized dynamics are assumed. Discussed are technical difficulties which are pertinent to the VSRA systems and have been overcome to achieve good model following capabilities. Two examples of VSRA's application to studying problems concerning to man-machine dynamic systems are included to show that the VSRA is a mandatory device to some classes of flight mechanical problem. The one is related to evaluating a newly proposed mode-decoupling (named Relaxed Static and Speed Stability -- RSS^2) system, and the other is to investigate pilot's capability for detecting failures in control system assuming an aircraft accident.

Introduction

An in-flight simulator was developed based upon a Beechcraft B-65 airplane at the National Aerospace Laboratory of Japan. The purpose of the in-flight simulator is to study a wide range of problems related to flying qualities, flight control systems and guidance law in terminal areas through flight experiments. To meet the purpose, the system must be provided with not only Variable Stability capability but variable Response capability, so that the Airplane is named "VSRA".

Ground based flight simulators are used for same purposes and there exists some criticism for utilizing an in-flight simulator as a general purpose test equipment. In this presentation, simple description of, and inherent limitations associated with, the VSRA, as well as major problems to obtain acceptable performance as a simulator, will be given. Also, to show that in-flight simulation is exclusively necessary to some class of flight-mechanical problems, presented are applications of VSRA to (1) evaluating a new mode-decoupling system named RSS^2 and to (2) studying pilot's ability to detect failures which occurred in flight control systems. The first application needs in-flight simulations because the system is essentially concerned with turbulence excitation both favorably and unfavorably. The second application is of simulating an unstable system and requests pilot's real tension for which in-flight simulation is mandatory.

1. VSRA System

Description

In designing the VSRA system, only linearized dynamics for rigid body are taken into account. An airborne computer stows a plural of prescribed mathematical model aircraft as well as necessary control laws. By these control laws, control commands to the plant are generated so that the selected plant outputs perfectly follow the model outputs which are responding to arbitrary control inputs applied by an evaluation pilot. The system was designed based upon an explicit model following theory[*1]. The control law is as shown in Fig.1.

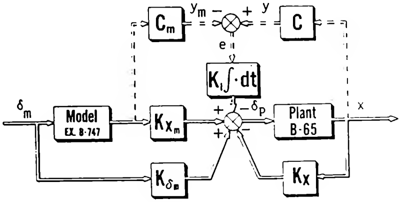


Fig.1 Control law

Although the system is described in some detail elsewhere[*2], a brief summary is included here for a quick reference. See Fig.2 for equipment arrangement. Pilot's seat is assigned for a safety pilot. Copilot's seat is assigned for an evaluation pilot with a mimic column, throttle, wheel and pedal. Direct lift device produces plus or minus 0.2 g incremental lift upto 2π(rad/s) sinusoid at 120 mph IAS. No manipulator is provided for the evaluation pilot to directly control the DLC. The DLC is driven by cross-feed signals from the mimic column and throttle inputs.

[*] Professor, Aerospace System Engineering, formerly National Aerospace Laboratory, Member AIAA

[†] Professor, Precision Machinary, formerly National Aerospace Laboratory, Member AIAA

[¶] Senior Researcher, Flight Research Division

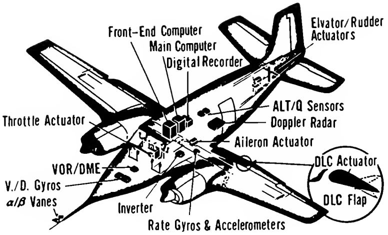


Fig.2 System arrangement of VSRA

All the mechanical linkages of control manipulators for the safety pilot are disconnected and instead, displacements of control input applied by evaluation pilot are picked up by position transmitters. Its increment δ_m from trimmed values is transmitted to the computer after necessary signal processing. A commercial mini-computer (3-MIPS) generates model states and control commands. Electro-mechanical servos in each control channels are driven by the commands and produce control deflections δ_p. Servo dynamics of each channels including surface inertia are well described by second order systems having cutoff frequency of 2π(rad/s). Computer cycle time Δ=1/25 (s), yields about 7.2 deg phase lag at the servo cutoff frequency.

In addition to the VSRA system, an onboard glide path computer (GPC) was developed in order to facilitate flight tests for simulated instrument approach. The GPC system utilizes medium accuracy sensors such as accelerometers, gyros, air data transducers and DME-VOR, and computes course deviations from a prescribed nominal approach path. As a byproduct, the GPC produces on-line estimates of gust components which the airplane is actually encountering. Fig.3 shows the schematic blocks of the GPC.

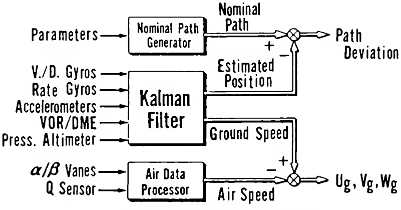


Fig.3 Glide path computer

Control law

Details of model following theory which the VSRA control system is based upon is included in ref.1. In order to discuss practical problems associated with choosing feedback gains, let us review its outline at first. Given are linearized system equations of the plant(VSRA),

$$\dot{x}_p = A_p x_p + B_p \delta_p \;;\; x_{p0} \tag{1a}$$

$$y_p = C_p x_p + D_p \delta_p \tag{1b}$$

and of a model,

$$\dot{x} = A_m x_m + B_m \delta_m \;;\; x_{m0} \tag{2a}$$

$$y_m = C_m x_m + D_m \delta_m \tag{2b}$$

respectively. Control law is given by

$$\delta_p = -K_{xp} x_p + (K_{xm} x_m + K_{\delta m} \delta_m). \tag{3}$$

By choosing a set of parameters which specify desirable output error dynamics, feedback gain K_{xp}, feedforward gain K_{xm}, and direct gain $K_{\delta m}$, are uniquely determined, so that plant output y_p follows model output y_m for arbitrary model input δ_m with the prescribed error dynamics; provided that (i) relative orders of each plant transfer function are not higher than corresponding relative orders of model transfer function, and that (ii) no transmission zeros of plant exist in the righthand s-plane.

Transmission zero is defined as the root of polynomial given by,

$$\Psi(s) = \begin{vmatrix} sI-A_p & -B_p \\ C_p & D_p \end{vmatrix} . \tag{4}$$

If no transmission zero of plant exists at all, poles of the output error dynamics coincide with those of the closed loop. Should any transmission zeros exist in the lefthand s-plane, they constitute a part of characteristic roots of error dynamics and only remainings can be arbitrarily assigned. Generally, one is given full or partial degree of freedom to choose error dynamics.

As is indicated in Appendix A , if the state variables (u,w,θ) are chosen as outputs to be matched in the longitudinal motion, and three control inputs $(\delta_e, \delta_t, \delta_f)$ are used, then one can assign the closed loop poles arbitrarily. When δ_f(DLC) is omitted and instead, only two outputs, for example, $(u,\ \gamma = \theta - w/U_0)$ are chosen, the same situation still holds under assumptions such as $X_{\delta e} = Z_{\delta e} = Z_q = 0$, which are acceptable from an engineering standpoint. As the other extreme, assuming a requirement to simulate effects of pilot station offset from aircraft center of gravity[*3],if all the pilot's acceleration cues i.e.,longitudinal and vertical linear accelerations as well as pitch angular acceleration are to be matched, one has no way to arbitrarily assign the convergence rate of output matching error because Ψ(s) of equation (4) includes 4(four) free s'. In this case, however, one can choose the integrated linear accelerations as well as the integrated attitude angle as the outputs to be matched. Again, no transmission zeros appear and output error dynamics are arbitrarily assignable.

In the lateral-directional motion where the VSRA is provided only two controllers, there is no essential problem in constituting control laws for model matching. It is shown in Appendix A that, for any choice of two out of (v,Φ,r) as outputs to be matched, desirable closed loop characteristics can be achieved. Under the assumptions $Y_p=Y_{\delta a}=Y_{\delta r}=0$ which may be accepted in most aircraft, lateral acceleration A_y is given by $Y_v v+Y_r r$ and the matching of A_y with mismatched reference speed (U_{0p} vs U_{0m}) is possible by appropriately choosing a row of C_p and C_m. Obviously, matching of three variables is not assured in this case.

Feedback gain

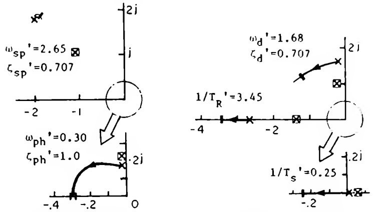
An important notice must be added here. Given an unstable model, response feedback type model following system can never reach a successful result. This is simply because the error dynamics is, and the model and the closed loop plant are too, unstable. As a general rule, to assure a quick convergence of model following error, all the assignable error dynamics should be stable enough and the feedback loop be tight enough. This requirement is diluted by various side constraints, among which of impotance are:

(i) Coupling of rigid body with servo dynamics As previously noted, servo dynamics in each control channel is best described by a second order dynamic system ($\omega_c=2\pi$ rad/s, $\zeta_c=0.7$). Too high feedback gains easily deteriorate the inherent stability of either servo or rigid body dynamics. The situations are absolutely undesirable.

(ii) Nonlinearity in thrust control Due to nonlinearity in throttle /manifold pressure/thrust relation, throttle control is better excluded from feedback loop.

(iii) Gust response of plant with loop closed Responses to atmospheric turbulence give a most challenging problem to in-flight simulation. Aparting from detail discussions on the subject, a simple but crucial relation between longitudinal model following error and horizontal gust, should be pointed out. Assuming no transmission zero and no feedback loop closed, error dynamics are specified by the open loop poles. Generally, error dynamics corresponding to the open loop phugoid poles would never be tolerated because of their sluggishness and poor convergence. Using elevator control, one can augment the phugoid roots to an increased pair of ω'_{ph} and ζ'_{ph} not only by attitude closure but by feedback of u_a and w_a. A more positive M_u, however, augments pitch angle excitation due to horizontal gust U_g at frequencies around or higher than ω'_{sp}, and thus spoils pilot's realism to fly a model airplane.

Considering such constraints as stated above, compromises are unavoidable. Fig.4(a) shows desirable pole locations of typical longitudinal closed loop(i.e. of output error) dynamics. Characteristic roots corresponding to phugoid are allocated on the negative real axis with acceptable magnitude of ω'_{ph} and within acceptable limit of M_u. Fig.4(b) also shows a desirable pole location for lateral-directional closed loop, where output errors in v and Φ are designed to decouple each other.

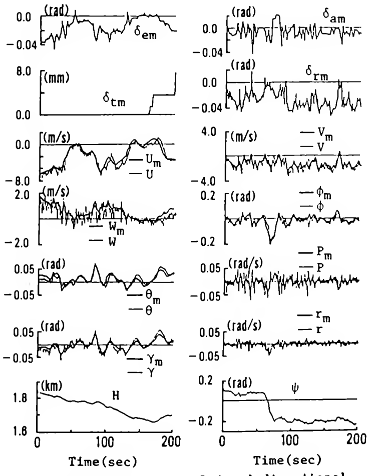


(a) Longitudinal (b) Lateral- Directional

Fig.4 Assigned error dynamics (✕ :Open-loop pole ╂:Closed-loop pole ☒ :Model pole)

2. System Evaluation

Fig.5 shows typical flight test results of both longitudinal and lateral-directional axes. The model is a domestic 35 ton transport in approach configuration @ U_{0m}=120 mph and γ_{0m}= -3 deg. The evaluation pilot was asked to make a shallow descent while keeping a heading with some doglegs. Transfer functions $(u,\gamma)/(\delta_e,\delta_t)$ and $(v,\Phi)/(\delta_a,\delta_r)$ are matched. In Fig.4, poles of open-loop plant and of model are compared with those of assigned error dynamics. Table 1 summarizes the numerators of



Longitudinal Lateral-directional

Fig.5 Evaluation of model following capability

longitudinal mode (high frequency zeros are omitted.). Although non-minimum zero exists in some numerators not only of model but of plant, perfect model-following capability is demonstrated as the theory predicts. In sequels, potential causes of model following error will be discussed. Only the cases with positive relative orders (strictly proper, $D_p=D_m=0$) are considered hereafter.

Table 1 Numerators of model and plant

	Model	Plant
u/δ_e(m/s/rad)	121(s/1.73+1)	541(s/1,09+1)
γ/δ_e(rad/rad)	.05(-s/.004+1)	-1.6(s/.028+1)
$u/(\tau/W)$(m/s/--)	-3.8(-s/.018+1)	-131(-s/.44+1)
$\gamma/(\tau/W)$(rad/--)	$1.0((s/1.2)^2+2(.88s/1.21)+1)$	1.4(s/.1+1)

Model Following Error due to Off-Trim

The theory is based upon the linearized dynamics about a trimmed state. To get state and control variables, all the measured quantities are subtracted with their 'trimmed values' which are supposed to be the values at a perfectly trimmed condition. In actual flight test, there remains some off-trim value, $(\dot{x}_p)_{trim}= A_p(x_p)_{trim} + B_p(\delta_p)_{trim}$ which is an unknown constant and yields a matching errors $e_y = C_p x_{p1}$ where x_{p1} is the response of the closed loop to the step input $(\dot{x}_p)_{trim}$ with an initial condition x_{p10}. The VSRA is provided with an auto-trim mode for easy-to-trim by safety pilot. The auto-trim mode is just one of stored models which includes a stable feedback loop with a zero K_{xm} and diagonal direct gain $K_{\delta m}$. In the example shown in Fig.5, no significant model following errors of this class are observed.

Plant Parameter Identification

Assume that the gains K_{xp}, K_{xm} and $K_{\delta m}$ are designed based on systems (A_p,B_p,C_p) and (A_m,B_m,C_m), but the actual plant has a system (A'_p,B'_p,C'_p) which is slightly different from that is assumed, then the matching error due to system description error $(\Delta A_p,\Delta B_p,\Delta C_p)$, where $\Delta A_p = A'_p-A_p$ etc. is given by $e_y= C_p x_{p2}+ \Delta C_p x_p$ where

$$\dot{x}_{p2}= (A_p-B_pK_{xp})x_{p2}+ d_f \quad ; \; x_{p20}=0 \tag{5a}$$

$$d_f=(\Delta A_p - \Delta B_p K_{xp})x_p+ \Delta B_p (K_{xm}x_m+K_{\delta m}\delta_m). \tag{5b}$$

The matching errors of this category comprises not only plant states but closed loop response to the model states and of model inputs, or in other words, the errors include higher order convolutions of model inputs δ_m. The dependency of model following errors on δ_m is fatal to in-flight simulations, and therefore precise informations of system matrices (A_p,B_p,C_p) are mandatory.

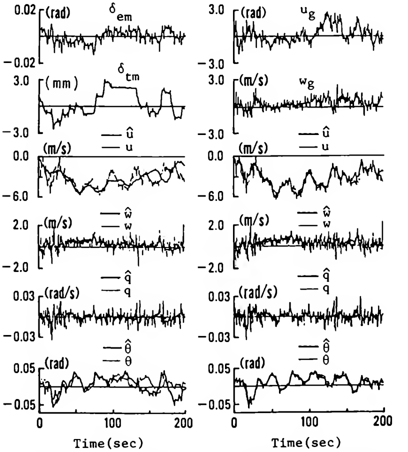
With these as motivations, frequency response method has been applied to identify control and stability derivatives, where a pure sinusoidal input is applied to each control axis at one time. Comparing with modern identification methods such as least squares or maximum likelihood, the authentic frequency response method is much more insensitive to unknown effects due to atmospheric turbulence and sensor biases and much easier to be recovered from filtering effects included in sensor-filter-recorder system either intentionally or unavoidably. The reliability of results is assured by checking compatibilities between two sets of estimate which are obtained by independently driving elevator or throttle and aileron or rudder, respectively (see Appendix B).

Effect of Atmospheric Turbulence

Assuming a model airplane that is flying in calm wind, existing atmospheric turbulence produces most typical model following errors, which are nothing but responses of the closed loop dynamics of plant to gust. Denoting as $x_{ap}= x_p+G\xi$ to accentuate that state variables (u,v,w) are defined as "relative to airmass", matching errors due to turbulence are given by $e_y= C_p x_{p3}$ where

$$\dot{x}_{p3}= (A_p-B_pK_{xp})x_{p3}+ G\dot{\xi} \quad ; \; x_{p30}= 0 \tag{5}$$

where if the point approximation*4 is assumed,



Computed response to control inputs only

Computed response to control and gust inputs

Fig.6 Gust estimation using GPC
(( ):measured,(^):computed)

$$G=\begin{pmatrix}-1 & 0 & 0 & 0\\ 0 & -1 & 0 & 0\end{pmatrix}^T,\ \xi=(u_g,w_g)^T\text{: longitudinal} \quad (7a)$$

$$=(-1\ \ 0\ \ 0\ \ 0)^T,\ \xi=v_g\text{ : lateral-directional.}(7b)$$

In order to show the possibility to eliminate the matching error of this kind, in Fig.6 is shown a typical result of gust detection using GPC. In the figure, the left column indicates control inputs δ_{em} and δ_{tm}, and the resulting control responses. Measured responses(thin lines) are compared with computed responses(bold lines (^)) to those particular inputs. Matching errors are obvious in both u and θ. In the right column, there are shown the turbulence (U_g,W_g) detected by GPC, and the computed responses thereof are added to the computed control responses and compared again with the measured. An acceptable matching is obtained between the computed and measured, which in turn proves that both the detected gust components and the estimated derivatives are sufficiently valid. It must be added here that the success of gust estimation owes adequacy in statistical modeling of DME noise spectrum in Kalman filter formulations. The doppler radar is only used for DME noise calibration but not for gust detection itself. The results of Fig.6 promises a possibility to suppress the undesirable gust response using only feed-forward loop.

3.Application to Evaluating RSS^2 System

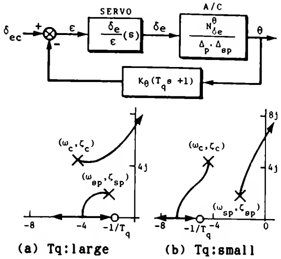
Decoupling of Pitch Control

It is said that most of the pilot ratings appear to be primarily determined by how precisely the pilot can control the airplane's pitch attitude*5. Two motives seem to exist for 'high gain pitch attitude closure'.

The one comes from the necessity to decouple the pitch mode. Precise control of attitude is said to be essential to establish and hold flight path or airspeed, and in fact, pitch attitude control are to be used as a command reference for path and speed control*6. To attain this situation, pitch control should be decoupled from other modes, whichever by pilot or control system. The need of sufficient decoupling calls for a θ-closure with high feedback gains. It should be noted here that the gains are often higher than those required for assuring just necessary bandwidth of pitch control.

The other originates from the relaxed static stability requirement. The term 'superaugmented' is defined*7 to be applied to aircraft that (1) are statically unstable without augmentation; (2) have a degree of pitch attitude stability with respect to inertial space; and (3) have pilot command/ aircraft pitch response characteristics that are largely independent of the aerodynamic stability derivatives. For such aircraft, 'high gain θ-closure' is necessary to cancel short period divergence and subsidence with pitch attitude/ elevator zeros.

As is well known, however, high gain closure of pitch attitude into elevator tends to result in a serious instability due to coupling between servo and short period mode. Fig.7 elucidates how the root loci tend to become unstable when the servo dynamics has a low bandwidth (phugoid roots are omitted). This problem stems from an essential lack of total system damping in pitch control dynamics*8 and the only cure would be to use a high bandwidth controller that is expensive. The high bandwidth controller requirements are said to be central problems related to superaugmented aircraft*7.


SERVO
A/C
δ_{ec}
+
-
ε
$\frac{\delta_e}{\varepsilon}(s)$
δe
$\frac{N^{\theta}_{\delta e}}{\Delta_p \cdot \Delta_{sp}}$
θ
$K_\theta(T_q s+1)$
(ω_c,ζ_c)
(ω_{sp},ζ_{sp})
4j
-8
-4
$-1/T_q$
8j
(ω_c,ζ_c)
4j
(ω_{sp},ζ_{sp})
-8
$-1/T_q$
-4
0

(a) Tq:large (b) Tq:small

Fig.7 Coupling of servo dynamics

Proposal of RSS^2 System

Being motivated with a desire to get an equivalently decoupled pitch attitude control with moderate feedback gains, the present authors proposed*9 a new mode-decoupling control law, that is given by

$$\delta_e = -(K_u u_a+K_w w_a+K_\tau \tau)- K_\theta(T_q q+\theta) + K_e\Delta_e \quad (8a)$$

where τ is thrust variation and Δ_e is column movement. If the spectra of u_a, w_a and τ are reasonably lower than the bandwidth of servo dynamics, by choosing gains such that,

$$(K_u\ \ K_w\ \ K_\tau)=(M_u\ \ M_w\ \ M_\tau)/M_{\delta e}$$
$$(8b)$$
$$K_\theta = (M_\theta+ \omega_\theta^2)/M_{\delta e},\quad K_\theta T_q=(M_q+2\zeta_\theta\omega_\theta)/M_{\delta e}$$

and referring to equation (A1), the longitudinal equation of motion (1) is decoupled into two modes, and actually it is residualized*10. A slow (airspeed and flight path) mode is given by,

$$\begin{bmatrix}\dot{u}_a\\ \dot{\gamma}\end{bmatrix}=\begin{bmatrix}X_u & -X_\alpha & u_a\\ -Z_u/U_0 & Z_w & \gamma\end{bmatrix}+\begin{bmatrix}X_\alpha-g & X_\tau\\ g\gamma_0/U_0-Z_w & -Z_\tau/U_0\end{bmatrix}\begin{bmatrix}\theta\\ \tau\end{bmatrix}$$
$$+\begin{bmatrix}-1 & X_w\\ 0 & Z_w/U_0\end{bmatrix}\begin{bmatrix}\dot{U}_g\\ W_g\end{bmatrix} \quad (9a)$$

and a fast mode is described by,

$$\begin{bmatrix} \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} -2\zeta_\theta\omega_\theta & -\omega_\theta^2 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} q \\ \theta \end{bmatrix} + \begin{bmatrix} K_e M_{\delta e} \\ 0 \end{bmatrix} \Delta_e . \qquad (9b)$$

In deriving equation (9a), the effect of Z_q is neglected for simplicity. Including Z_q yields an equation which is partly not strictly proper but no essential change will occur in sequels. Also in the derivations, the air speed u_a is retained as it is while the vertical airspeed w_a is converted into the inertial flight path angle using a relation of $\gamma=\theta-(w_a+w_g)/U_0$. The variables u_a and γ are considered to be of primary importance in an approach flight phase.

Fast Mode

Equation (9b) gives the pitch attitude response to column input

$$\frac{\theta}{\Delta_e} = \frac{M_{\delta e}K_e}{s^2+2\zeta_\theta\omega_\theta s+\omega_\theta^2} \qquad (10a)$$

where from equation (8b), one has

$$\omega_\theta^2=-M_\theta+ M_{\delta e}K_\theta, \quad 2\zeta_\theta\omega_\theta= -M_q+ M_{\delta e}K_\theta T_q. \qquad (10b)$$

Inspection of equations (9) and (10) reveals distinguished features of the system :

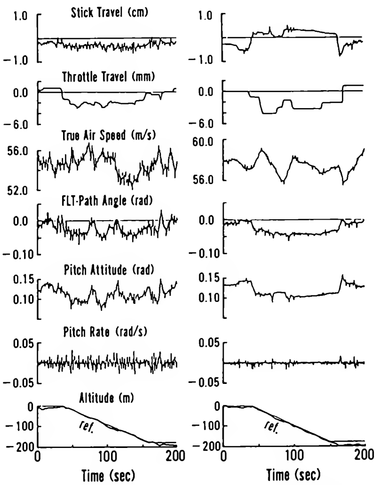
The original target of pitch mode decoupling is accomplished by nullifying moment derivatives M_u, M_w and M_T. Since both speed stability M_u and static stability M_w have been effectively relaxed, this system is named (RSS^2) system. By these moment nullifications, the pitch dynamics are, as in the case of 'high gain θ-closure', completely free from aerodynamic effects except for $M_{\delta e}$ and hence, from atmospheric disturbances.

The feedback gains T_q and K_θ must be solely assigned from the requirement to realize necessary bandwidth and damping of the augmented pitch mode. Since the gains are free from the requirement to mode decoupling, they are of moderate magnitude. The requirement of only finite gains contrasts with current 'high gain feedback'. In fact, one can design the system without being annoyed at possible coupling between servo and rigid body dynamics even using servos of a moderate bandwidth.

Speaking to aircraft pitch control, pilot's work load is expected to be reduced because of a flat pitch response to column input with assignable bandwidth and of low level (theoretically zero) response to turbulence.

Flight Test Results

Fig.8 compares flight test results of the bare frame configuration of B-65 with that of RSS^2 based attitude command/attitude hold configuration. During the tests, the evaluation pilot was asked to make a simulated instrument approach while keeping a constant air speed. Both tests were conducted consecutively in the same test area with the same approach direction, so that about the same level of atmospheric turbulence existed for both test cases. Drastic reductions both in pitch rate response and in high frequency column movement are clearly observed with the RSS^2 system engaged.



Bare frame(B-65) Bare frame + RSS^2

Fig.8 Effects of RSS^2

Slow Mode

The residualized slow mode is a point mass representation of motion comprising airspeed and flight path as state variables and pitch attitude θ and thrust variation τ as control commands. It is well known that characteristic roots of equation (9a) are zero of pitch/elevator transfer function, i.e. $s=-1/T_{\theta 1}$ and $s=-1/T_{\theta 2}$, which are determined by purely aerodynamic derivatives X_u, X_w, Z_u and Z_w, and located close to the origin. This situation is a natural consequence of pitch mode decoupling and is valid independently on the feedback gains (K_θ, T_q). In this sense, the RSS^2 gives a limiting case of θ-closure with infinitely high feedback gains.

Although the RSS^2 system is originally devised in VSRA program where rather low bandwidth servos were available, it seems to have potentially wider applications. In relaxed static stability aircraft, the derivative M_w gives insufficient stability by its definition, and there remains little reason to rely on M_u for keeping inherent stabilities. To the contrary, positive nullification of it would be preferable. As a basic control system for conventional aircraft, RSS^2 offers a potential benefits as described above. For such applications, gain scheduling, or less possibly, some adaptive feature becomes necessary to implement equation (8).

Both the RSS^2 and high gain θ-closure systems possess a potential drawback by nature. In Fig.8, observed are excursions in airspeed which follow the glide slope capture and leveling off. The transfer function matrix $(u_a,\gamma)/(\theta,\tau)^T$ of equation (9a) has a form $A_{lm}(s+1/T_{lm})/\Delta(s)$, where $l=u_a$ or γ, $m=\theta$ or τ and $\Delta(s)=(s+1/T_{\theta 1})(s+1/T_{\theta 2})$. Gains A_{lm} and time constants T_{lm} are given in ref.6. For the VSRA and other conventional aircraft without power effect, numerator zeros $1/T_{u\theta}$ and $1/T_{u\tau}$ approximately cancel the $s = -1/T_{\theta 2}$ mode, and thus leaves $s= -1/T_{\theta 1}$ as the speed mode. Since the control law (8a) requires feedback loop to elevator, transfer function numerators relating to column input are invariant to the loop closure. On the contrary, numerators relative to thrust input are significantly affected by the loop closure.

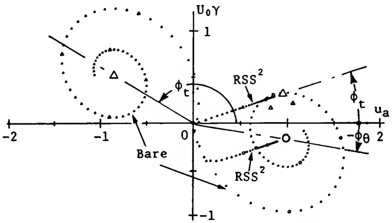


Fig.9 Step response in u-γ plane

In Fig.9, the step responses of both u_a and γ to column and thrust inputs are compared for bare frame and with RSS^2 system engaged. The magnitude of step inputs Δ_e and δ_t is so normalized that $(u_a)^2+(U_0\gamma)^2=1$. The vectors pointing the steady states indicate (re-trimmable) nonzero set point by each particular control. Let the angle of steady state vector measured from positive u_a axis be denoted by Φ_θ (step θ input) and Φ_t(step τ input), respectively. If these two steady vectors align each other ($\Phi_t - \Phi_\theta = 0$ or π), one cannot re-trim the aircraft to arbitrary (u_a,γ) point but can only to a point along these vectors. It is easy to show that $\tan(-\Phi_\theta)$ is proportional to the backside parameter $1/T_{\gamma\theta}$, and hence $|\Phi_\theta|$ is close to zero in operation at around the bottom of power required curve.

With RSS^2 system engaged, pitch attitude is held constant even when thrust is increased,and hence at steady state, flight path angle changes as much as the change in angle of attack. The shares of change in angle of attack and in airspeed due to thrust increase are determined by purely aerodynamic relations. From equation (9a), obtained is

$$\begin{bmatrix} X_u & -X_w \\ -Z_u & Z_w \end{bmatrix}\begin{bmatrix} u_a \\ U_0\gamma \end{bmatrix} + \begin{bmatrix} X_\tau \\ -Z_\tau \end{bmatrix}\tau = 0 .$$

Thus, as extreme cases, when $|Z_\tau| << X_\tau$

$$\tan(\Phi_t) = (\frac{U_0\cdot\gamma}{u_a})_{ss} = Z_u/Z_w = 2C_L/C_{L\alpha} \qquad (12a)$$

and when $|Z_\tau| >> X_\tau$

$$= X_u/X_w = -2C_D/(C_L - C_{D\alpha}) \qquad (12b)$$

Equations (12) give a rough idea how drastically the direction of responses to thrust changes with the RSS^{2}closed or 'high gain θ-closure'. As is the case of VSRA, most of conventional CTOL aircraft has Φ_t of less than 10 degrees and undesirable condition $\Phi_t \doteqdot \Phi_\theta$ comes out. Associated problems with such control cross coupling has been discussed in connection with powered lift aircraft[*11] .

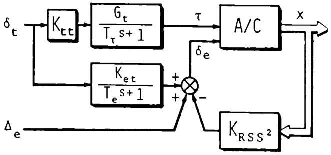


Fig.10 Cross feed of throttle command

The VSRA is readily used to investigate this control cross coupling problem. By cross feeding throttle command with equivalent thrust lag $1/T_e$ into attitude command as is shown in Fig.10 , arbitrary thrust responses direction Φ_t can be simulated. Flight tests[*12] are conducted with various combinations of Φ_t and the relative

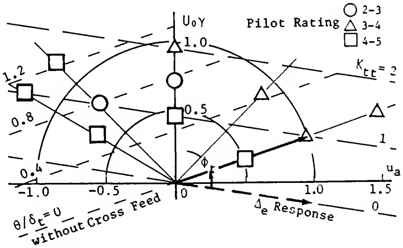


Fig.11 Simulated steady state thrust response direction and corresponding pilot ratings

magnitude of thrust response vector at steady state (throttle effectiveness). Evaluation pilots are requested to make a simulated instrument approach. Obtained pilots' ratings are annotated in Fig.11. Taking pilots' comments into account, followings are observed from the limited number of data; (i) there exists an desirable throttle effectiveness i.e. too low or too high effectiveness yield low ratings. (ii) for less than 45 degrees of $(\Phi_t-\Phi_\theta)$, ratings deteriorate and between 45 to 135 degrees, flat ratings are given. The analyses need more elaboration but it must be understood that the apparent drawback of RSS^2 can be covered by a simple cross feed.

4.Application to Failure Detection in Flight Control Systems by Pilot

Several aircraft accidents are reported wherein a part of control surfaces or control systems has failed[*13,*14]. Assuming that minimum flyablility is retained after such failures, properly detecting the mode of failures is mandatory to survive. Despite of provision for various failure annunciators and warning devices, it is unlikely that the pilots rightly and timely detect[*] the total mode of failures. Such a lack of detectability may be attributable to the defect of monitoring and warning systems currently under use. Failures actually occurred are often those which have hardly been hypothesized or modelled a priori, so that the pilots were never given true aspects of failure by a single or any combinations of warning annunciation. In fact, a number of warning lights turned on at a time gives no information. On the other hand, pilots in themselves must be an ultimately intelligent monitoring/detecting element as well as a controller in a man-vehicle system.

As a part of the accident investigation[*14], simulation tests using VSRA were conducted to study pilots' detectability to partial loss of aerodynamic surfaces and failures in augmented flight control systems[*15]. Only concerned is the part of pilot's detectabilities which are embodied through their efforts in manually controlling a failed aircraft.

The normal and failed configurations are defined in Table 2. Due to disengagement of autothrottle and autopilot at the time of initial failure, both longitudinal and lateral-directional stability characteristics are changed from the augmented to the bare. Due to partial loss of vertical fin and total loss of rudder, dutch-roll mode becomes unstable. The corresponding characteristic roots are depicted in Fig.12. At some time later of initial failure, control effectiveness of elevators and ailerons are lost due to low hydraulic pressure.

Table 2 Definition of normal and failed modes

Before failure (normal mode):	
* Airframe	normal
* All hydraulic systems	normal
* Yaw damper	on
* Pitch and roll stabilization	on
After failure (failed mode):	
* Part of vertical fin	lost
* Total rudder	lost
* Yaw damper	off
* Pitch and roll stabilization	off
* Horizontal stab. & elevator	inoperative
* Aileron and spoiler (after some time lag)	inoperative

A plural of mathematical models are stored in the airborne computer. The system operator switches, at arbitrary moment at his option, models from the normal one to the failed one with aileron control effectiveness, then to the one with aileron control lost leaving only thrust variation as a positive control. In some test cases, artificial random disturbances are added to excite rolling motion of few degrees in root-mean-square sense.

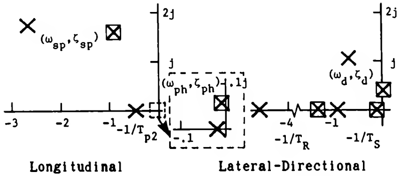


Fig.12 Characteristics roots before ✕ and after ☒ failure

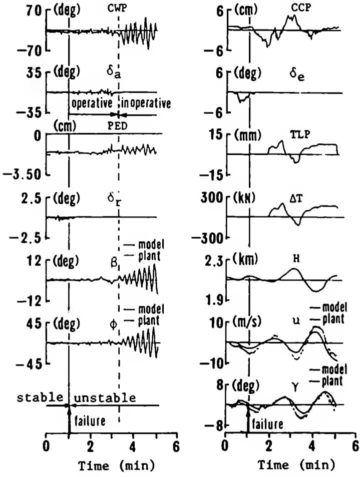


Fig.13 Simulation of control system failure and its detection by pilots

[*] Failure detection consists of three tasks[*16]: (1) 'alarm' is to make a binary decision, either something goes wrong or everything is fine, (2) 'isolation' is determining the source of the failure, (3) 'estimation' is to determine the extent of the failure.

Evaluation pilots are asked to maintain a straight and level flight without clear outside visuals, and to report anything unusual whenever he gets 'alarmed'. He is also requested to 'isolate' and if possible, to 'estimate' the mode of failures. Fig.13 shows a typical time history before and after simulated failures. In this case, pitch and roll autopilot and yaw damper systems as well as rudder and elevator controls are cutoff at t=1 min (initial failure), and the aileron control is brought into failure at slightly after t=3 min. With very low level atmospheric turbulence and without artificial roll disturbance, the evaluation pilot isolated failure of elevator at 28 sec and of aileron at 25 sec after each events. After he is alarmed with elevator failure, he tries to stabilize phugoid mode by throttle control with bad success. The lateral-directional modes are excited by slightly unbalanced thrust between LH and RH engines. The time for detection are of average from test case to case and from pilot to pilot. Generally, pilots never fail to report the loss of control effectiveness in elevator or aileron after some detection time.

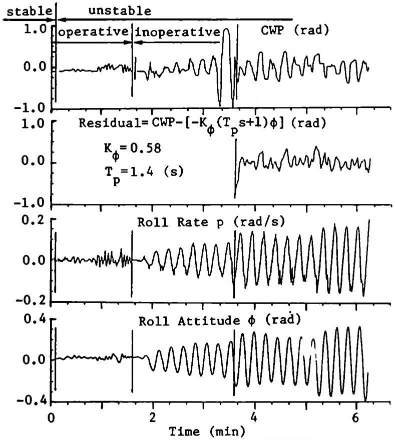


Fig.14 Estimation of pilot's feedback gains with and without aileron control effectiveness

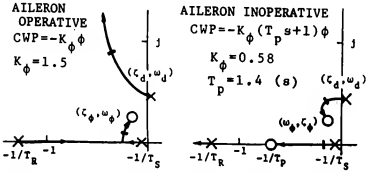


Fig.15 Estimated root loci with and without aileron control effectiveness

To the contrary, they scarcely isolated the failures in stability augmentation systems(e.g. wings leveller) as far as the pertinent control (e.g. aileron) remains alive. Fig.14 shows another example of lateral-directional time history. It is seen that after the loss of stability but before the loss of aileron control, the pilot well manages to suppress dutch roll oscillation by using aileron control. Assuming that the pilot closes the loop by proportionally feeding back the bank angle into aileron, a root locus is plotted in Fig.15. Judging from the dominant frequency evident in roll rate time history in Fig.14, which is about .15 to .2 Hz, pilot's feedback gain K_ϕ of about 1.5 deg/deg is estimated. It is presumed that because the pilot successfully stabilizes the unstable airplane using such a lower feedback gain or, with less workload, even he himself does not recognized the loss of stability. It shall be stressed here that observing the state of airplane such as roll angle from outside of the man-vehicle closed loop, one can be given no chance at all to detect any abnormality as far as the loop is well stabilized.

Least squares estimate of pilot's feedback gain after having lost the aileron control effectiveness, gives a proportional and differentiation feedback law as is shown in Fig.15 with its root locus plot. Despite of inclusion of lead term in his feedback loop which requests him some extra work load, the dutch roll oscillation still persists or even diverges. This outcome should be disagreeable enough to the pilot, so he detected the failure definitely.

In Fig.14, there is seen an obvious indication of a kind of test signal in pilot's CWS position at around t=3.3 min. In this particular test case, artificial disturbance in roll axis is added, and the pilot detects aileron control loss rather earlier (within about 5 sec) than the cases with no artificial disturbance. However, after his first detection, the pilot became not confident of his initial conclusion that the control has completely lost. This is because of confusing the roll responses induced by disturbance with those due to his own control. As a result of his test signal input, he recovered confidence again.

Conclusions

(1) The VSRA in-flight simulator's system and associated limitations are described.
(2) Potential sources of model following error are discussed from practical stand point. Of those described in some details are: (i) uncertainty in system description of plant, for which frequency response method gives exclusively adequate methodology, (ii) offtrim (offset in trim) condition, for which an auto-trim mode gives a solution, and (iii) atmospheric turbulence, the closed loop response of which gives major source of model following error and its effect is most possibly suppressed by an feedforward gust alleviation system based upon on-line estimates of gust component.
(3) As an application of VSRA to investigate flight mechanical problems, a new mode-decoupling system named RSS^2 which attains exactly the same target that the infinitely high gain Θ-closure aims and

which needs only finite feedback gains. Flight test results demonstrating the merit of the system is included. A possible drawback which is common to RSS^2 and high gain θ-closure system is pointed out, wherein response of airspeed and flight path angle to thrust input is greatly changed. Flight test results are included to study the control cross coupling problem and to prove that crossfeed from throttle to elevator copes with the problem.

(4) As the second example of VSRA's application, flight simulation tests for investigating pilot's detectability of failures which occurred in flight control systems are presented. The test program was conducted to prepare background data for a recent aircraft accident investigation and a mechanism of failure detection by pilot who is manually controlling a failed airplane is obtained.

VSRA in-flight simulator has certain difficulties as a general purpose test device, such as lack in all weather capability, poor mimic manipulators, speed limitations and so forth. It has proved to be extremely useful for some class of flight mechanical problems where realism such as pilot's tension or atmospheric turbulence is of primary importance.

References

*1 Kawahata,N.,"Model-Following System with Assignable Error Dynamics and Its Application to Aircraft,"Journal of Guidance and Control,Vol.3, No.6,1980,pp.508-516

*2 Komoda,M.,Kawahata,N.,Tsukano,Y.,and Ono,T., "Variable Stability and Response Airplane,Part I and Part II," Japan society for Aeronautics and Space Sciences,Vol.31,No.349, 1983(in Japanese)

*3 Weingarten,N.C., and Chalk,C.R., "In-Flight Investigation of Large Airplane Flying Qualities for Approach and Landing,"Journal of Guidance, Control, and Dynamics",Vol.7,No.1, 1984, pp.92-98

*4 Etkin,B.,Dynamics of Atmospheric Flight, John Wiley & Sons,Inc., New York, 1972 pp.544-547

*5 Neal,T.P.,and Smith,R.E.,"A Flying Qualities Criterion for the Design of Fighter Flight-Control Systems," Journal of Aircraft, Vol.8,No.10, Oct. 1971, pp.803-80

*6 Franklin,J.A., and Innis,R.C.," Flight-Path and Airspeed Control during Landing Approach for Powered-Lift Aircraft," NASA TN D-7719, Oct.1974

*7 McRuer,D., Johnston,D., and Myers,T.," A Perspective on Superaugmented Flight Control: Advantages and Problems," Journal of Guidance, Control and Dynamics, Vol.9,No.5, 1986, pp.530-540

*8 McRuer,D. ,Ashkenas,I., and Graham,D., Aircraft Dynamics and Automatic Control,Princeton University Press,Princeton, N.J., 1973

*9 Kawahata,N. and Komoda,M., "Mode Decoupling by Attitude Control with Emphasis to Power Effect and Gust Response," Proceedings of 21st Aircraft Symposium, JSASS,Kyoto,Japan,1983 (in Japanese)

*10 Stengel,R.F., "A Unifying Framework for Longitudinal Flying Qualities Criteria," Journal of Guidance, Control, and Dynamics, Vol.6,No.2,1983, pp.84-90

*11 Heffley,R.K., Stapleford,R.L.,and Rumold,R.C., "Airworthiness Criteria Development for Powered-Lift Aircraft," NASA CR-2791,1977

*12 Kawahata,N., Komoda,M., Tsukano,Y., Ono,T., Inagaki,T., and Ishikawa,K., "Improvement of RSS^2 Characteristics by Cross-Feed and Flight Evaluation," Proceedings of 25th Aircraft Symposium, JSASS,Tokyo, Japan,1987 (in Japanese)

*13 NTSB Buraeu of Accident Investigation: Aircraft Accident Report--- China Airline Boeing 747-SP, N4522V, Repot No. NTSB/AAR-86/03

*14 Aircraft Accident Investigation Report--- JAL 747SR, JA8119, Report No. 62-2(in Japanese)

*15 Komoda,M., and Kawahata,N.,"Failure Detection in Flight Control Systems by Pilots," Preprint,40th International Air Safety Seminar, Flight Safety Foundation,Inc.,Tokyo, Oct. 1987

*16 Willsky,A.S., "A Survey of Design Method for Failure Detection in Dynamic Systems," Automatica, Vol.12, 1976, pp.601-611

Appendix A

Longitudinal Axes

Assuming sufficiently high bandwidth response of elevator, throttle(manifold pressure) and flap, the mathematical representation of plant in stability axes is given as follows.

$$x_p=(u,w,q,\theta)^T, \qquad \delta_p=(\delta_e,\delta_t,\delta_f)^T \tag{A1}$$

$$A_p=\begin{bmatrix} X_u & X_w & 0 & -g \\ Z_u & Z_w & U_0+Z_q & -g\gamma_0 \\ M_u & M_w & M_q & M_\theta \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{A2}$$

$$B_p=\begin{bmatrix} X_{\delta e} & X_{\delta t} & X_{\delta f} \\ Z_{\delta e} & Z_{\delta t} & Z_{\delta f} \\ M_{\delta e} & M_{\delta t} & M_{\delta f} \\ 0 & 0 & 0 \end{bmatrix} \tag{A3}$$

In equations (A2) and (A3), original derivatives $Z^*_{\dot w}$ and $M^*_{\dot w}$ are lumped with inertial terms and swept out to obtain a state space form. Hence, denoting with * original derivatives as defined in the literature[*8], $Z_{u/w}=Z^*_{u/w}/(1-Z^*_{\dot w})$, $M_{u/w}=M^*_{u/w}+M^*_{\dot w}Z_{u/w}$, $Z_q=(Z^*_q+U_0Z^*_{\dot w})/(1-Z^*_{\dot w})$, $M_q=M^*_q+M^*_{\dot w}Z_q$, $\gamma_0=\gamma^*_0/(1-Z^*_{\dot w})$, and $M_\theta=-g\gamma_0M^*_{\dot w}$.

If all state variables are chosen as the outputs to be matched, $y=(u,w,\theta)^T$, then

$$C_p=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A4}$$

and equation (4) yields,

$$\Psi(s)=\begin{bmatrix} X_{\delta e} & X_{\delta t} & X_{\delta f} \\ Z_{\delta e} & Z_{\delta t} & Z_{\delta f} \\ M_{\delta e} & M_{\delta t} & M_{\delta f} \end{bmatrix} \tag{A5}$$

which is a nonzero constant, and no transmission zero exist in finite s-plane. On the contrary, if longitudinal and vertical accelerations A_x,A_z and angular acceleration $\dot{q}$ are chosen as outputs, then

$$C_p = \begin{bmatrix} s & 0 & 0 & g \\ 0 & s & -U_0 & g\gamma_0 \\ 0 & 0 & s & 0 \end{bmatrix} \tag{A6}$$

yields the same $\Psi(s)$ of (A5) with 4 free s' in addition. To avoid this difficulty, one should consider 'type 1' control. By choosing a set of following integrated quantities as outputs to be matched,

$$\begin{aligned} A_{x1} &= \int (A_x)dt = u + g\theta_1 \\ A_{z1} &= \int (A_z)dt = w - U_0\theta + g\gamma_0\theta_1 \\ \theta_1 &= \int (\theta)dt \end{aligned} \tag{A7}$$

and by augmenting the state x_p and the system matrices A_p and B_p using a kinematic equation $\dot{\theta}_1 = \theta$, then one has again equation (A4), that means the plant does not possess transmission zeros.

Lateral-Directional Axes

$$x_p = (v,p,\Phi,r)^T, \quad \delta_p = (\delta_a,\delta_r)^T \tag{A8}$$

$$A_p = \begin{bmatrix} Y_v & Y_p & g & -U_0+Y_r \\ L_v & L_p & L_\Phi & L_r \\ 0 & 1 & 0 & 0 \\ N_v & N_p & N_\Phi & N_r \end{bmatrix} \tag{A9}$$

$$B_p = \begin{bmatrix} Y_{\delta a} & Y_{\delta r} \\ L_{\delta a} & L_{\delta r} \\ 0 & 0 \\ N_{\delta a} & N_{\delta r} \end{bmatrix} \tag{A10}$$

Equations (A9) and (A10) originate from the primed derivatives[*8], and then the original derivatives $L^*_{\dot{v}}$ and $N^*_{\dot{v}}$ are lumped with pure inertial terms and swept out. Hence, for example, such terms as $L_\Phi = gL^*_{\dot{v}}$, $N_\Phi = gN^*_{\dot{v}}$ appear.

Since no side force generator is provided, only two controls are available and hence two outputs can be matched. Generally, under an acceptable assumption, $Y_p = Y_{\delta a} = Y_{\delta r} = 0$, the equation (4) yields

$$\Psi(s) = \Psi_1(s) \begin{bmatrix} L_{\delta a} & L_{\delta r} \\ N_{\delta a} & N_{\delta r} \end{bmatrix}. \tag{A11}$$

If $y=(v,r)^T$ or $y=(v,\Phi)^T$ are chosen as outputs to be matched, then $\Psi_1(s)=g$ or $\Psi_1(s)=U_0-Y_r$, respectively, so that no transmission zeros exist. If $y=(\Phi,r)^T$ are chosen, $\Psi_1(s) = s-Y_v$ so that one of error dynamics is fixed by $s=Y_v$, which is stable.

Appendix B[*2]

Consider identifying a system (A,B), $B=(b_1,b_2)$,

$$\dot{x} = Ax + b_1\delta_1 + b_2\delta_2 . \tag{B1}$$

The system is sinusoidally excited by both δ_1 and δ_2. Transfer functions are given by,

$$\begin{aligned} h_l(s) &= (sI-A)^{-1}b_l \\ &= n_l(s)/\Delta(s) \quad , \ l=1,2 \end{aligned} \tag{B2}$$

where $\Delta(s)$ is characteristic polynomial and $n_l(s)$ is a column of numerator polynomials to input δ_l. In order that two sets of estimate (A,b_1) and (A,b_2) be compatible, (i) estimates of $\Delta(s)$ must coincide when driven by either δ_1 or δ_2 . Putting $\Lambda=\mathrm{diag}(\lambda_i)$ with λ_i characteristic root, and $T=(x_1,x_2,\ldots,x_n)$ with x_i the corresponding normalized characteristic vector, $h_l(s)$ is written by

$$\begin{aligned} h_l(s) &= T(sI-\Lambda)^{-1}b^*_l \\ &= \left(\frac{x_1}{s-\lambda_1} \ \cdots \ \frac{x_i}{s-\lambda_i} \ \cdots \ \frac{x_n}{s-\lambda_n} \right) b^*_l \end{aligned} \tag{B3}$$

where $b^*_l = T^{-1}b_l$ and therefore Residue(h_l @ $s=\lambda_i$) = $(b^*_l)_i \cdot x_i$. But since Residue(h_l @ $s=\lambda_i$) is proportional to $n_l(s=\lambda_i)$, another compatibility is obtained, i.e. (ii) assuming all distinct characteristics roots of A, $n_1(\lambda_i) = k_i \cdot n_2(\lambda_i)$ must hold, where k_i is a real or complex constant.

88-4608-CP

NASA SHUTTLE TRAINING AIRCRAFT FLIGHT SIMULATION OVERVIEW

CHARLES R. JUSTIZ*
NASA/JOHNSON SPACE CENTER
HOUSTON, TEXAS

SURESH M. PATEL**
NASA/JOHNSON SPACE CENTER
HOUSTON, TEXAS

Abstract

The Shuttle Training Aircraft (STA) is a variable stability, variable control law flying simulator used by NASA/JSC to train astronauts in the final landing phase of a space shuttle orbiter. A general outline is given for the STA flight simulation system. This outline includes the basic algorithms of the simulation. An overview is given of the software generation and verification process through the Advanced Validation System (AVAS). The flight test techniques for software verification will be reviewed and the process for releasing the software for flight training will be covered. The astronaut STA training syllabus is examined. Parameter matching with the Orbiter in the final approach phase of de-orbit and landing is brierly examined. Simulation performance will be assessed against flight data, performance measurement, and cue synchronization. Model following techniques employed in the STA will be correlated to performance.

Acronyms and Nomenclature

γ	Inertial Flight Path Angle
δ	Flight Control Deflection
Θ,Ψ,Φ	Euler Angles
Ac,Ec,Rc	Aileron, and Elevator, Rudder Command
Ax,Ay,Az	Body Axis Accelerations
ADAS	Advanced Digital Avionics System
AT	Acceptance Test
AVAS	Advanced Validation System
AUTO	Automatic Guidance Command
BF	Orbiter Body Flap
CG	Center of Gravity
D	Drag
DADC	Digital Air Data Computer
DLC	Direct Lift Control
$\overline{e}_D$	Average Energy Dissipation Rate
err	Error Signal
FSE	Flight Simulation Engineer
g	gravity
GCC	Guidance & Control Computer
G-II	Gulfstream II
h	Estimated Altitude
HAC	Heading Alignment Cone
L	Lift
LRU	Line Replaceable Units
m	Model
MCDS	Multifunction Cathode-Ray Tube Display System
NEP	Nominal Energy Point
MLS	Microwave Landing System
p,q,r	Angular Body Rates
PFT	Pre-Flight Test
R	Range Distance
RHC	Rotational Hand Controller
SB	Speed Brake
STA	Shuttle Training Aircraft
STS	Space Transportation System
TAEM	Terminal Area Energy Management
Ve	Equivalent Airspeed
Vg	GroundSpeed
Vi	Inertial True Airspeed
W	Weight
X,Y,Z	Estimated Position Vectors

I. Basic Simulation Problem

The STA provides Orbiter pilots with a realistic simulation of Orbiter cockpit motion, cues, and handling qualities, while simultaneously matching the Orbiter's atmospheric decent trajectory from 35,000 feet to the actual Orbiter cockpit height above the runway (approximately 20 feet) at touchdown. This is accomplished through independent control of the aircraft in 6 degrees-of-freedom, using standard autopilot control of pitch, roll, and yaw supplemented by direct lift control (DLC) and in-flight reverse thrust. In the simulation mode, control of the DLC's and the in-flight reverse thrust, as well as the conventional aircraft controls, is performed by an onboard digital computer: the Advanced Digital Avionics System (ADAS).

The Orbiter's final approach and landing phase is simulated by the STA from 35,000' MSL to simulated touchdown. This portion requires the STA to be capable of simulating the guidance, navigation and control to intercept and follow the Heading Alignment Cone (HAC)(see Fig. 1).

* Research Pilot, Johnson Space Center
Senior Member AIAA

**Control Systems Engineer, Johnson Space Center

Upon rolling out aligned with the runway, the Orbiter would establish a two-phase final, the steep (outer) glide slope and the shallow (inner) glideslope. The steep glideslope nominally begins approximately 7nm from the runway threshold with the Orbiter approximately 12,000 ft. A.G.L., 19° gamma and 290 KEAS. The Orbiter will maintain this glideslope until 1750 AGL at which time it accomplishes a preflare maneuver which will transition it to a 1.5° glideslope, 300 ft AGL approximately 1 NM from the touchdown point. The nominal touchdown point is 2750 ft. from the threshold of the runway.

Simulation System Description

The six degree-of-freedom Orbiter model containing the aerodynamic equations-of-motions generates the responses due to the pilot's input and the automatic navigation control system inputs. Typical Orbiter responses are angular and translational accelerations, rates, and displacement. The model following control systems accomplishes the matching of the STA responses to the Orbiter responses generated by the six degree-of-freedom model.

The navigation software blends STA space positional information obtained from the various navigation aids with model vectors to obtain estimates used in the guidance laws. Steering commands which are displayed on Orbiter instruments at the Simulation Pilot's Station are coupled to the Orbiter control system model either through the pilot flying the various Orbiter control modes or in the AUTO mode.

Orbiter Guidance.

During simulation of the Orbiter atmospheric flight phase, the STA is, theoretically, an unpowered glide vehicle. The guidance and flight control system must control energy, flight path angle, and heading to align the STA (Orbiter) with the runway at an appropriate altitude and airspeed in order to perform an unpowered landing.

Because of the criticality of the Orbiter model energy state, both at the start and the termination of this flight phase, it is referred to as Terminal Area Energy Management (TAEM). This software contains the Optional TAEM Targeting (OTT) guidance which improves the Orbiters' energy dissipation required to interface with the approach and land (A/L) guidance system. OTT consists of four phases:

1. Acquisition. Straight in or overhead approach to a point of tangency on HAC.
2. Heading Alignment. Initiated at point of HAC tangency when the radial distance from the center of the HAC is less than the spiral turn radius multiplied by 1.1. End of phase when the predicted range is less than the computer prefinal initialization range or when altitude is less than 7000 feet.
3. Prefinal. Removes lateral dispersion and airspeed deviation. Delivers vehicle to A/L steep glide slope phase. A/L transition at altitude less than 10,000 feet (nominal) or 7000 feet (forced).
4. Steep Glide Slope. Maintains runway heading on the steep glide slope for transition onto the shallow glide slope. Nominal gamma equals -19 degrees (light weight) or -17 degrees (heavy weight).
5. Pull-Up. Flares from steep glide slope to capture shallow glide slope when altitude is less than 2000 feet.
6. Shallow Glide Slope. Maintains 1.5 degree glide slope by flying out slope errors down to an altitude less than 80 feet (nominal) or 30 feet (forced).
7. Final Flare. Maintains constant descent rate until two feet above runway; transitions onto runway at h' = -3 fps.

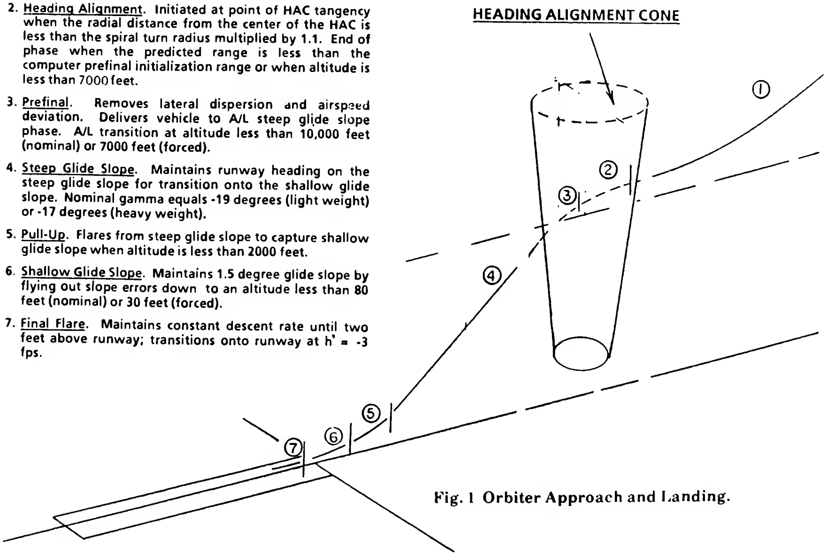


Fig. 1 Orbiter Approach and Landing.

- S-turn
- acquisition
- heading alignment
- prefinal

The S-turn logic is presently deactivated in the STA. The A/L guidance automatically acquires and maintains the vehicle on an approach trajectory from TAEM guidance termination to touchdown. If the STA state variables are maintained within tolerances, the unpowered landing can be executed either manually or automatically.

In an equilibrium glide state in which centrifugal acceleration is negligible, the following relationships between aerodynamic lift (L), drag (D), and weight (W) are applicable:

$$L = W * Cos\,\gamma \qquad (1)$$

$$D = \frac{L}{(L/D)} = \frac{W*Cos\,\gamma}{(L/D)} \qquad (2)$$

For unpowered flight, in transversing a range distance (R), the loss in energy is equal to the aerodynamic drag (D) times the distance transversed along the drag vector.

$$\Delta E = D*R = \frac{W*Cos\,\gamma}{L/D} * R \qquad (3)$$

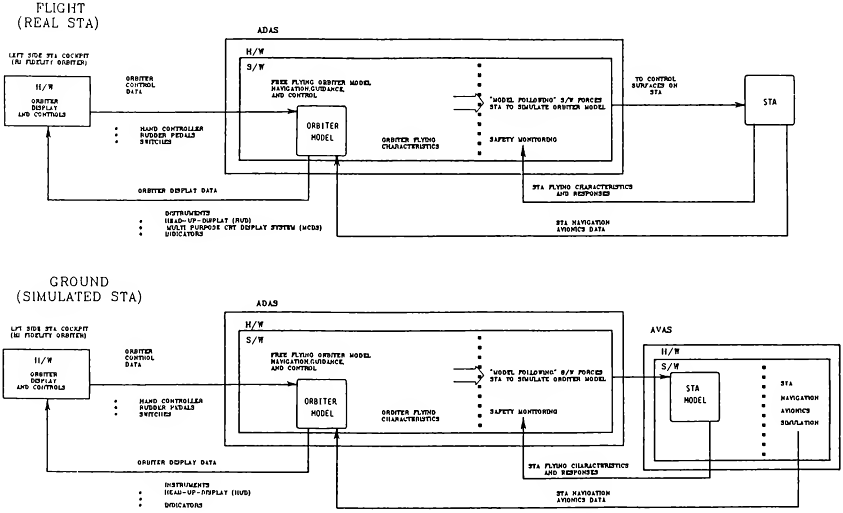
The glide range (R) is calculated from navigation space positional information and the vehicle total energy (potential plus kinetic) normalized with respect to weight is given by:

$$(E/W)\ Actual = (h + (V_T)^2/2g \qquad (4)$$

The amount of energy to be dissipated is the difference from the current state to that required at the Nominal Entry Point (NEP) for the approach and landing guidance. Dividing the difference by the glide range to the NEP yields the average energy dissipation rate e$_D$ which is inversely proportional to the lift-to-drag ratio.

$$\bar{e}_D = \frac{(E/W\,actual - (E/W\,desired)}{R} = \frac{\Delta(E/W)}{R} = \frac{1}{(L/D)} \qquad (5)$$

Basically, there are two types of approaches to the runway. A straight in or an overhead approach may be made to the HAC, which is used to align the vehicle with the runway. The ground track of an overhead approach around the HAC will be a spiral. When the vehicle is in


FLIGHT (REAL STA)

GROUND (SIMULATED STA)

ADAS
H/W
S/W
LEFT SIDE STA COCKPIT (HI FIDELITY ORBITER)
H/W
ORBITER DISPLAY AND CONTROLS
ORBITER CONTROL DATA
HAND CONTROLLER
RUDDER PEDALS
SWITCHES
FREE FLYING ORBITER MODEL NAVIGATION,GUIDANCE, AND CONTROL
ORBITER MODEL
ORBITER FLYING CHARACTERISTICS
"MODEL FOLLOWING" S/W FORCES STA TO SIMULATE ORBITER MODEL
SAFETY MONITORING
TO CONTROL SURFACES ON STA
STA
ORBITER DISPLAY DATA
STA FLYING CHARACTERISTICS AND RESPONSES
INSTRUMENTS
HEAD-UP-DISPLAY (HUD)
MULTI PURPOSE CRT DISPLAY SYSTEM (MCDS)
INDICATORS
STA NAVIGATION AVIONICS DATA
AVAS
STA MODEL
STA NAVIGATION AVIONICS SIMULATION

Fig. 2 STA Flight and Ground Configuration.

the heading alignment phase of TAEM and the turn angle is greater than 90 degrees, the final radius of the spiral may be adjusted between a maximum of 14,000 feet to a minimum of 5,000 feet for extreme low-energy situations. The radius of turn for the spiral is calculated based on the final radius of the spiral and the HAC turn angle. The spiral will vary in size depending on the amount of HAC turn angle required.

Model Following Control System.

In order to force the STA to respond to Rotational Hand Control (RHC) commands in the same manner as the Orbiter, the Advanced Digital Avionics System employs a modified implicit model following scheme. Generally, the computed response of the Orbiter is compared with the measured STA flight responses and the difference is used to drive the STA control surfaces. Orbiter model "feed-forward" (FF) is used to reduce the lag between the Orbiter response and the STA response. These are generally Orbiter model flight control parameters that allow STA control laws to anticipate Orbiter response. The following is a discussion of each STA control law.

A. Elevator Control Law. The error between the computed Orbiter pitch rate and the measured STA pitch rate plus the error between the computed Orbiter pitch attitude and the measured STA pitch attitude is used to drive the STA elevator control law. The Orbiter pitch attitude is biased low because the STA cannot attain the high angles-of-attack that the Orbiter can. For a given Orbiter weight option, this theta bias is constant. A cockpit window screen used during astronaut training will account for the bias angle as well as for the window size differences between the STA and Orbiter. Orbiter model elevator command, pitch acceleration, and angle-of-attack rate (alpha dot) are the elevator feed-forwards used. There is also a pitch rate compensator in the elevator control law that is not being used in the nominal configuration. The compensator is designed to improve the STA frequency response in comparison to the Orbiter frequency response.

The elevator trim tab runs automatically to off-load the elevator

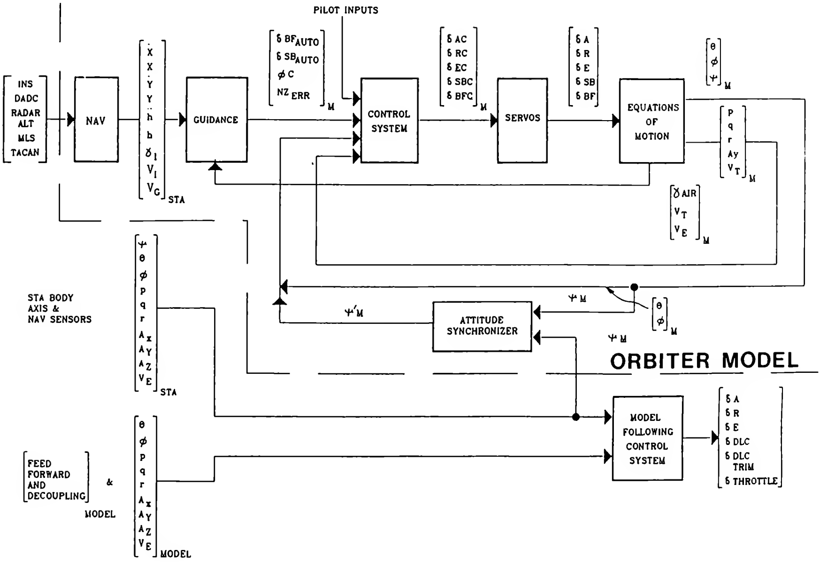


Fig. 3 STA Model Following Block Diagram.

servo. The servo current is used to command trim tab motion.

B. Direct Lift Control (DLC) Law. The error between the computed STA cockpit acceleration plus the error between the computed Orbiter flightpath angle and the computed STA flightpath angle is used to drive the DLC control surfaces. The flaperons and flaps respond to the same error signals, only the flaperons move more slowly and have a large deadband. The flaperons move simultaneously in response to DLC commands and differentially in response to aileron commands. There is also an Orbiter normal acceleration feed-forward that improves Nz model-following.

C. Aileron Control Law. The error between the computed Orbiter roll rate and the measured STA roll rate plus the error between the computed Orbiter roll attitude and the measured STA roll attitude is used to drive the aileron control law. The flaperons move differentially in response to aileron commands. The ailerons trim tab runs automatically to off-load the aileron servo. Servo current is used to command trim tab motion.

D. Autothrottle Control Law. The error between the computed Orbiter true airspeed and the measured STA true airspeed plus the error between the computed Orbiter longitudinal acceleration and the measured STA longitudinal acceleration is used to drive the STA throttle. The throttles are used in reverse thrust during trajectories.

E. Rudder Control Law. The error between the computed Orbiter yaw rate and the measured STA yaw rate plus a combination of roll rate and aileron command terms (for turn coordination) is used to drive the rudder control law. It is important to note that the STA rudder authority is limited to 1.5 degrees through the series servo in order to prevent the rudder from exciting a "vertical fin-rocking mode." This limit prevents the STA from following the Orbiter properly when large Orbiter yaw rates are commanded.

Model Following Performance

The time history plots of the orbiter and the STA flight responses to standard orbiter step response (OSR) inputs are presented in Figures 4 through 6.

The orbiter and the STA pitch rate (Fig. 4) show that the STA response lags the orbiter by 0.2 seconds. This lag is introduced by the transport delay through the system and by the STA flight control system lag.

The vertical acceleration plots (Fig. 5) show that the STA response has almost zero delay. This is because of an additional lead-lag filter introduced in the feed forward loop of the Az command. Also, the DLC servo response of 50 degree/second is much higher than that for the other servos in the STA.

The roll rate response (Fig. 6) shows that the STA lags the orbiter by 0.2 seconds.

II. Software Generation and Verification

The STA system is basically divided in three primary catagories:

- Advance Digital Avionics System (ADAS)
- Advance Validation System (AVAS)
- Pre Flight Test (PFT)

These software modules are composed of real-time and non real-time systems at all language levels, specifically, assembly, high order and operating system. The ADAS and the AVAS are the real-time systems, with the ADAS executing at a cycle time of 50 millisecs and the AVAS executing at a cycle time of 40 milisecs. The PFT is a non real-time system.

Advance Digital Avionic System (ADAS)

This on board flight system is incorporated in one "black box" refered to as Guidance Control Computer (GCC) on board the STA. The GCC actually consists of two separate computers. The Sperry SDP-118A computer simulates the six degrees of freedom Orbiter equations of motion, Orbiter navigation, guidance and control, model following, safety monitoring and various mode logics. This computer is programmed in it's own specialized Assembly language and contains more than 35000 lines of code

The input/output (I/O) processor is a MC68010 microprocessor based computer which is programmed in Assembly language. The function of this processor is to control power-up and reset sequences, to process the input output

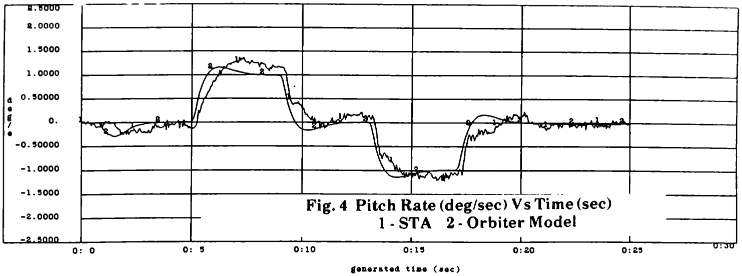


Fig. 4 Pitch Rate (deg/sec) Vs Time (sec)
1 - STA 2 - Orbiter Model

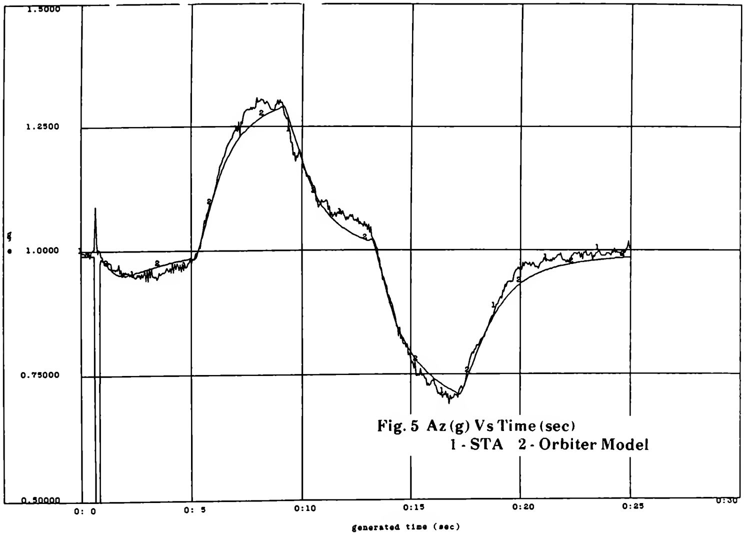


Fig. 5 Az (g) Vs Time (sec)
1 - STA 2 - Orbiter Model

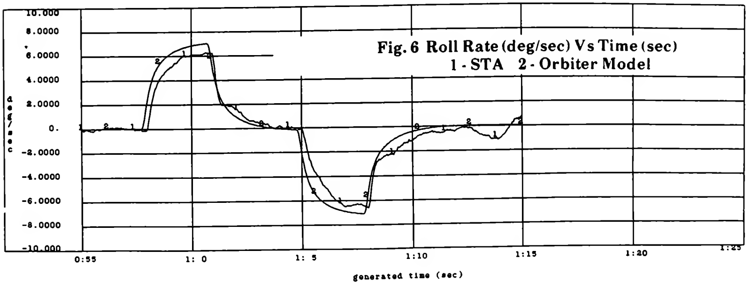


Fig. 6 Roll Rate (deg/sec) Vs Time (sec)
1 - STA 2 - Orbiter Model

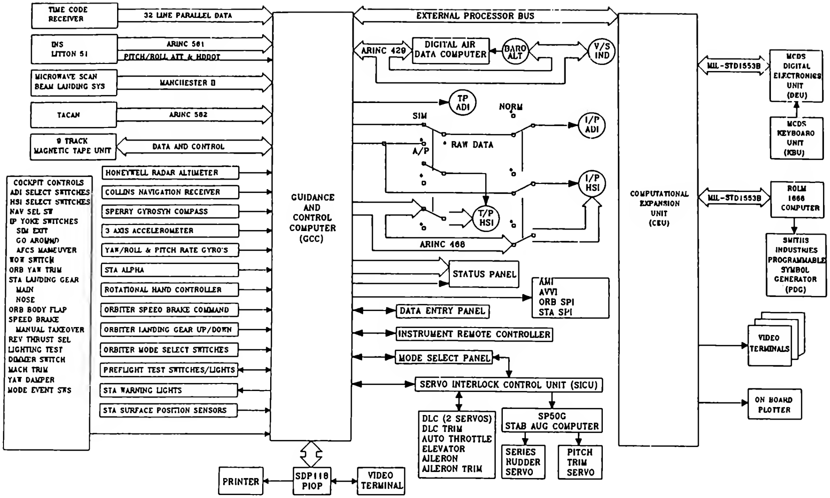


Fig. 7 STA Avionics Overview.

data, record data on magnetic tape, generate heart beat to continually monitor the health of the GCC and the process failure actions and reporting tasks.

The computational Expansion Unit (CEU) is also a MC68010 microprocessor based computer which provides data interface for the Heads Up Display (HUD) and the Multifunction CRT Display Syustem (MCDS).

Advance Validation System (AVAS)

The AVAS is the fixed based simulation of the six degree-of-freedom equations of motion of the basic G II airframe as well as the STA airborne sensors, instrumentation and input/output signal processing. This is a MC68010 based system with programming done in Assembly and "C" languages. The AVAS is used for testing the line replacable units (LRU) and development and testing of the flight software

Pre Flight Test (PFT)

The PFT is a specialized software loaded in the STA GCC for conducting a thorough pre-flight check of the airplane instrumentation, flight control system, sensors, switches and mode engage logics. This is a non-real time system providing prompts and cues to the flight simulation engineer (FSE) and then verifying the response to the FSE action.

Software Generation Process

In general, the STA system is a complex network of seven central processors programmed in six different assembly languages and two different high level languages. The Space Shuttle pilot's training and the safety of the flight depends heavily on the quality of the software, and therefore, a systematic process with stringent quality control, has been set for software generation and verification.

The software requirements are normally generated by the research pilots, FSE's or system engineers. These requirements normally reflect the Space Transportation System (STS) initialization loads changes, improvements in flight monitoring and safety and engineering developments. These requirements are thoroughly reviewed by engineering, flight operations, quality assurance and flight safety. The software designer, will then design the software updates to implement these requirements into the existing flight software. A requirement may call for changes in more than one processor. The software designs are once again thoroughly reviewed to ensure that no inadvertent changes are made and that no hidden failures exist. Once the software coding is completed, the software is turned over to the requirement originators for testing and evaluation.

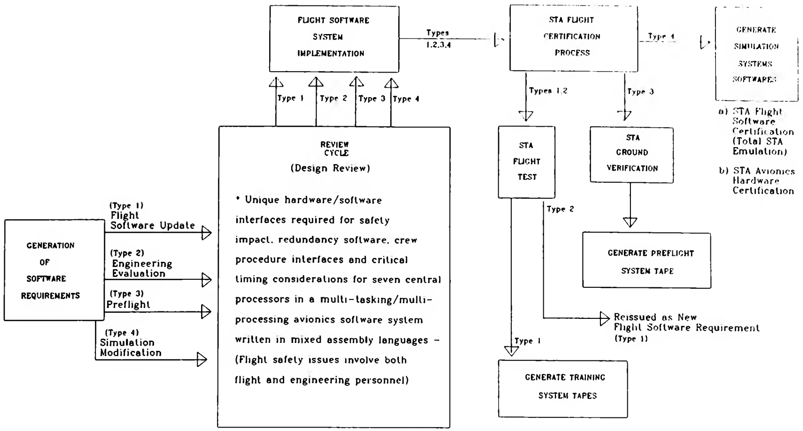

FLIGHT SOFTWARE SYSTEM IMPLEMENTATION
Types 1,2,3,4
STA FLIGHT CERTIFICATION PROCESS
Type 4
GENERATE SIMULATION SYSTEMS SOFTWARES
a) STA Flight Software Certification (Total STA Emulation)
b) STA Avionics Hardware Certification
Type 1
Type 2
Type 3
Type 4
Types 1,2
Type 3
REVIEW CYCLE
(Design Review)
* Unique hardware/software interfaces required for safety impact, redundancy software, crew procedure interfaces and critical timing considerations for seven central processors in a multi-tasking/multi-processing avionics software system written in mixed assembly languages – (Flight safety issues involve both flight and engineering personnel)
STA FLIGHT TEST
STA GROUND VERIFICATION
GENERATION OF SOFTWARE REQUIREMENTS
(Type 1) Flight Software Update
(Type 2) Engineering Evaluation
(Type 3) Preflight
(Type 4) Simulation Modification
Type 2
GENERATE PREFLIGHT SYSTEM TAPE
Reissued as New Flight Software Requirement (Type 1)
Type 1
GENERATE TRAINING SYSTEM TAPES

Fig. 8 STA Software Development Process.

Software Verification Process

During the engineering evaluation of the updated software, the engineer develops a test plan to validate his or her requirements and to ensure a safe software update. All these test requirements are once again reviewed by associated personnel, including the quality assurance representative. The final on ground Flight Software Acceptance Test (AT) is conducted. The AT includes a base line test which is common to all software deliveries. This test ensures that the overall simulation performance is not significantly different from the previous software delivery. The new software changes are then tested.

The AT data are recorded on strip chart recorders, XY plotter and on magnetic tape for off line data reduction. Any discrepancies or unacceptable occurances during the AT are reviewed in detail and if no reasonable explanation exists, further AT is stopped. After completion of the AT, the test data are reviewed and the updated software is handed over to the flight operation for further flight tests.

III. Flight Test

Test Maneuver Descriptions

The following systems flight tests are carried out for new software programs that are developed for crew training on the STA.

1. Orbiter Step Routine (OSR) Ramps - Any software change which modifies or affects the flight control system or model following performance requires verification of command/response performance by testing with OSR ramps. The OSR ramp is introduced directly into the affected control system(s). In general, the OSR can consist of RHC command amplitude of 1.5° to 12° control deflection, has a rise time of 1 second, a hold time of 2 seconds, and a settle time of 1 second.

These tests are done at both 190 KEAS and 290 KEAS.

2. Frequency Response - Frequency response testing is not conducted on a routine basis as a part of the verification of STA software modifications. However, changes which affect the Orbiter model or model following, and therefore the frequency response matching between the STA model and the Orbiter model, must be evaluated by limited frequency response testing.

The test conditions for frequency response are nominally:

Altitude	-	15,000 ft MSL
Airspeed	-	190 KEAS or 290 KEAS
RHC Command Amplitude		± 5° or ±10° (sinusoidal)
Frequency		0.1 Hz to 2.0 Hz

Aircraft Handling Qualities Evaluation - The assessment of aircraft performance which results from software modifications must be done to a significant extent by handling qualities evaluation by the flight crew. The following areas are evaluated by flight crews in verifying the flightworthiness of a new software delivery for training. Although no quantitative evaluation is utilized, flight crew assessment is based on aircraft performance differences between new and existing software programs.

a. Horizontal Model Following - The areas which are to be evaluated are:

(1) Short Period Response - Assessment of damping and residual oscillations ("ringing") characteristics during RHC singlet and doublet inputs.

(2) Stability in Maneuvering Flight - Assessment of rate-hold performance while executing approximately 40° banked turns at both low speed and high speed.

(3) Lateral/Directional Response - Execute bank-to-bank and bank-to-a-point maneuvers to assess lateral acceleration, adverse yaw, and roll/sideslip coupling characteristics.

(4) DLC (Direct Lift Control) System - Assess DLC system performance by transitioning from low airspeed to outer glide slope speeds.

(5) Qualitative TMF Evaluation - general TMF performance is assessed during maneuvers which require the pilot to do aggressive capture and hold guidance tracking.

IV. STA Training Requirements -

A minimum of 500 STA approaches is the desired experience level for pilots prior to their first Shuttle mission. The basic STA syllabus provides approximately 240 approaches. The pilot is eligible for flight assignment upon completion of the basic STA Syllabus. Shuttle crew assignments occur approximately one year prior to flight. STA flying after crew selection provides an additional 270 approaches. From one year to nine months prior to launch, the pilot receives one sortie per month (10 approaches per sortie). From nine months prior to launch to three months prior to launch, the pilot receives two sorties per month. When the pilot is within three months of launch, he receives a sortie each week.

V. Conclusion -

The STA is a mature flying simulator system. Responses received from astronauts following their Orbiter missions indicates that the STA performs with excellent fidelity and that the approach and landing training is indispensible. Further improvements are constantly being evaluated in both the hardware and software of the system to include single higher level language system (ADA), processor upgrades (from MC68010 to MC68020), cockpit upgrades (LCD glass cockpit) as well as many others. Engineering modifications to improve the model following performance, flight safety, and flight operations are also constantly in work.

88-4609-CP

GROUND SIMULATOR REQUIREMENTS BASED ON IN-FLIGHT SIMULATION

Louis H. Knotts* and Randall E. Bailey**
Calspan Advanced Technology Center
Buffalo, NY

Abstract

Several recent NT-33A in-flight simulation projects have addressed issues relevant to ground simulator fidelity. During two of these studies a comparison was made of handling qualities for several aircraft configurations when flown in the NT-33A compared to the same configurations flown in a ground simulator. Piloting tasks consisted of visual landings and head-up display tracking tasks. During one of these studies systematic variation of added time delay was made for several generic types of aircraft in the flight as well as the ground simulator environment. Observations were made concerning the effects of piloting task, simulator motion, and time delay on aircraft handling qualities. A third study consisted of an in-flight investigation into the effects of feel system dynamics and time delay on lateral handling qualities. It was found that low frequency artificial feel systems can significantly degrade handling qualities. The findings of this study can be applied to control loader requirements for ground simulators. A common theme of the generic simulation studies performed in the NT-33A is that calibration and documentation is an essential step in the set-up of a simulation. This is true not only for simulation of aircraft dynamics, but also for other characteristics which may affect handling qualities such as time delay and control stick characteristics.

Introduction

The complexity and expense of modern aircraft have made the use of ground simulators increasingly important not only for the design of these new aircraft but for the training of their crew. With regard to new aircraft development, ground simulators allow for piloted e·aluations of many control modes tailored to a specific mission or to a unique aircraft configuration. Ground simulations for aircraft development must be sufficiently accurate in their representation of handling qualities to allow evaluation of candidate flight control systems. In a training simulator the objective of the simulation is the adequate preparation of aircrews for flight without the use of an expensive aircraft or an unnecessarily complex ground simulator. In both uses of simulation, for aircraft development and for training, it is important to know what characteristics of the simulation provide the necessary simulation fidelity.

Several recent NT-33A research programs have addressed issues relevant to flight simulation fidelity:

- In 1985 an experiment was conducted to compare the handling qualities of aircraft configurations which are prone to pilot induced oscillations (PIO's) in both a real aircraft and a ground simulator. The NT-33A was used as the flight test vehicle while the NASA Ames Vertical Motion Simulator (VMS) was programmed as the companion ground simulator. The ground simulator phase of the study, therefore, contained large amplitude motion and a wide field-of-view CGI visual system. The NT-33A provided real world visual and motion cues during an approach and landing task.
- The second NT-33A program which was designed as a comparison study between ground and flight simulation was conducted in several phases between 1986 and 1988. This project utilized the NT-33A as both the flight test vehicle as well as a fixed base ground simulator. The visual system in both cases was restricted to Head-Up Display (HUD) symbology. The objective of the study was to determine allowable time delays for several different aircraft types during up-and-away tracking tasks.
- The third NT-33A flight evaluation program of relevance to simulator fidelity consisted of a flight investigation into the effect of feel system dynamics on aircraft handling qualities. The conclusions of this study can be applied to the artificial feel system requirements for a ground simulator.

The results of these three NT-33A projects, the significance of time delays and feel system dynamics, and the importance of documenting such characteristics in a ground simulator are discussed in this paper.

NT-33A In-flight Simulator

The NT-33A in-flight simulator aircraft (Figure 1) is a highly modified Lockheed jet trainer owned by the Air Force Flight Dynamics Laboratory and operated under contract by Calspan Corporation [1]. The front cockpit set of aircraft controls has been replaced by a full authority fly-

Figure 1 USAF NT-33A In-flight Simulator

[*] Principal Engineer, Member AIAA
[**] Senior Engineer

by-wire flight control system and a variable response artificial feel system. The dynamic characteristics of the aircraft can be varied in-flight by means of an analog response feedback variable stability system as well as an onboard multipurpose digital computer. The evaluation pilot, who sits in the front cockpit, controls the simulated aircraft through a standard centerstick and rudder pedal arrangement or a sidestick controller. A fully programmable HUD is installed in the front cockpit. The rear cockpit of the NT-33A contains the original mechanical flight control system which provides a back-up to the experimental front seat system. A safety pilot occupies the rear cockpit where he controls normal aircraft systems, the in-flight experiments, and provides a safety monitor for the evaluation pilot.

Time Delays

Before the results of the NT-33A studies into simulation fidelity can be described in detail, a perspective should be gained regarding time delays and their effect on aircraft handling qualities. Any delay between a pilot's command input and the onset of an airplane's motion response can have a significant effect on the pilot's ability to control an aircraft. This is true for real aircraft in flight, and is especially true for ground simulators. In fact when pilots state that they have to learn to "fly" a ground simulator they often mean that they are modifying their control techniques so as to compensate for time delays. In an airplane, time delays are generated by: the inherent delay in the onset of motion due to the aerodynamics; the additional delay in initial response due to dynamic elements such as the control stick, flight control system components, or surface actuators; and pure delays which may be added by the flight control computer. These different types of delays are illustrated on Figure 2 which shows a first order response with no delay; the same first order response with .2 seconds of "pure" delay; and the first order system cascaded with nine high frequency filters to produce a 10th order system containing a large "equivalent" or "effective" delay.

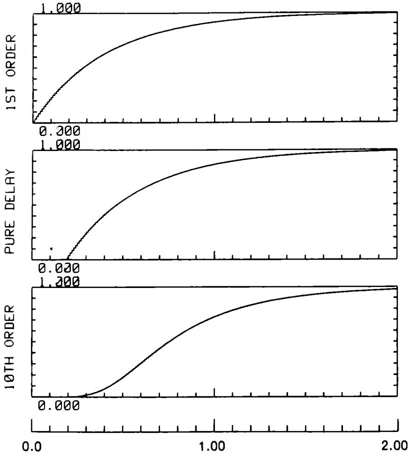


Figure 2 **First-order system with no delay; with pure delay; and 10th order system with equivalent delay**

An artificial feel system can contribute to the overall time delay of an aircraft. If a flight control system uses stick position as the command input from the pilot, then the dynamics of the feel system as it converts the pilot's force input to a stick position are an integral part of the aircraft's command path. Accordingly, the stick dynamics, particularly if the stick is low frequency, contribute to the throughput time delay. For a force command flight control system, stick force is used as the command input. The dynamics of the stick and the resulting displacements of the stick are of no significance to the flight control system with regard to time delay, although as will be discussed later, they have a significant contribution to the closed-loop handling qualities of the vehicle. The relationship of position and force command flight control structures to the closed loop pilot/aircraft system is shown on Figure 3.

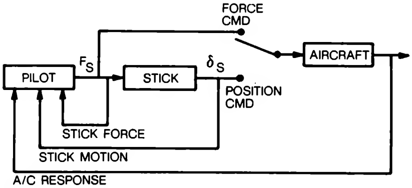


Figure 3 **Contribution of the stick to the closed loop system for both force and position command flight control systems**

In addition to real aircraft delays, a ground simulator adds further delay in the onset of a pilot's sensory cues because of the need to calculate the motion of the vehicle, and to present that motion to the pilot through motion drive systems and artificial scene generation[2]. It is often these additional delays in the ground simulator which cause the pilot to fly the simulation differently than he would the aircraft. Since delays are a problem which must be evaluated during the development process of an aircraft, these additional delays can exasperate an accurate assessment of a new flight control design during a development program when using a ground simulator.

Much work has been done during the past decade to identify acceptable limits of time delay in aircraft based on handling qualities considerations[3,4,5] . This has not been a simple task because an acceptable level of time delay depends on aircraft dynamics, the piloting task at hand, and individual pilot technique. With regard to task, a small change in the task definition in the presence of time delay can cause the handling qualities of an aircraft to change from acceptable to dangerous. This phenomenon is known as a handling qualities cliff. An example of this was seen when the Space Shuttle was first landed on a runway rather than a dry lake bed[6]. Although acceptable flying qualities were evident when the less-precise lakebed landings were made, a significant pilot induced oscillation (PIO) developed when an exact touchdown point was attempted on the runway . Since that time, the shuttle pilots have learned to adapt to the presence of time delays by using a low gain "pulse and wait" control strategy. This specialized technique is similar to what pilots often do in ground simulators to adapt to the presence of time delays. During a landing task this technique often requires the pilots to hold aircraft attitude and wait for touchdown during the final portion of a landing. This type of piloting technique has the disadvantage of being "open loop" and not exposing differences between candidate control system designs during an aircraft development program. With regard to training simulation, this control technique may be different from the control strategy which must be used in the real aircraft and a negative transfer of training results. In summary, the presence of time delays can significantly alter flying qualities; a ground simulator always has more time delay than the aircraft it simulates; and how much added delay is too much is not precisely known.

Ground and In-flight Comparisons

Two recent NT-33A projects were designed specifically to compare the handling qualities of aircraft configurations in-flight and in a ground simulator with various levels of time delay. These experiments were designed to produce piloted evaluations using aircraft models with the same dynamic response and stick characteristics, pilots, evaluation tasks, and experimental procedures for both the ground and in-flight phases of the study. Little data has been found from previous experiments which have placed these same constraints on ground simulation versus flight comparisons.

The first NT-33A simulation comparison study was sponsored by the NASA Ames Research Center and flown at Edwards AFB in December 1985[7]. The test matrix included five longitudinal aircraft configurations with handling qualities which ranged from good (a rating of 2 on the Cooper-Harper scale[8]) to unflyable (CHR=10). The variations in flying qualities were produced primarily by adding time delay to an otherwise good aircraft—pure delays of 100 milliseconds and 144 milliseconds, and an equivalent delay of about 117 milliseconds due to a 12 r/s second-order prefilter. The unflyable airplane was generated by adding a low frequency 4th order prefilter and, consequently a great deal of equivalent time delay, to a reasonably good aircraft. The evaluation task consisted of a visual landing task in which the pilot lined up with the edge of the runway until 100 feet above the ground. He then corrected to line up on the runway centerline and attempted to touchdown at a precise point. This lateral offset maneuver has been used on many in-flight simulation programs in order to force the pilot to use high pilot gain throughout the landing task.

The handling quality results received in the NT-33A for the five configurations were in general as anticipated. However, the pilots tended to fly more open loop as the program progressed. This resulted in better handling quality ratings for a given time delay during later flights. For example, one pilot rated the configuration with 100 milliseconds of added delay a CHR=5 on his first flight. On his fourth flight the same configuration was rated a CHR=3. Another pilot rated the 144 milliseconds of added delay configuration as a CHR=6 on his third flight, while on his fifth ride this configuration was a CHR=3. This trend points out the importance of designing an evaluation task that requires the pilot to stay in the control loop and sample the aircraft's handling characteristics. As the pilots in this study learned to perform the offset correction task, they became more predictive and cautious of extraneous or spontaneous control inputs. To avoid this tendency, a properly designed flight experiment would inject random disturbance inputs or instruct the pilots to try less than optimum setups for the landings. During the experiment in question, these task variations were discouraged because an exact duplication of the landing task was to be repeated during the ground simulator phase of the study.

The worst aircraft configuration received Cooper-Harper ratings of 8,9, and 10 during the NT-33A flights by two of the three evaluators. In the VMS, these pilots gave the same configuration ratings of 5 and 6 using the original lateral offset landing task. A second, more demanding task was then tried in the ground simulator. The nominal lateral offset visual approach was flown. After the line-up correction was made at 100 feet, a cockpit light was randomly illuminated at 25 feet which indicated that the pilots were required to extend their touchdown point 500 feet further down the runway. When the worst configuration was evaluated in the ground simulator using this task, ratings of 9 and 10 were given. The ratings of the better configurations did not change when the more severe landing task was performed. A summary of the pilot ratings for the five configurations in both the NT-33A and VMS is provided in Table 1.

Table 1 Summary of Cooper-Harper ratings for NT-33/NASA VMS simulation comparison study

EVALUATION PILOT

Aircraft Configuration	A			B			C	
	NT-33	VMS Task 1	VMS Task 2	NT-33	VMS Task 1	VMS Task 2	NT-33	VMS Task 1
1	4, 2	4, 4	4, 3	3, 5, 3	3, 3	3	4, 3, 3	3.5, 4
2	3, 2, 2	5, 6	5, 5	3, 4	3, 4	3	5, 3	4, 5
3	2, 3	7, 4	5, 7	6, 3	3, 4	5	6, 7	3.5, 4
4	2, 2	5, 6	6, 4	3, 3	4	2, 5	5, 2	4.5, 3
5	6,7	7.5, 10	9, 10	10, 9	5, 5	10, 9	9, 8	6, 5.5

The second NT-33A program which was designed as a comparison study between ground and flight simulation was sponsored jointly by the Air Force Human Resources Laboratory (HRL), Aeronautical Systems Division (ASD), Armstrong Aerospace Medical Research Laboratory (AAMRL), Flight Dynamics Laboratory (FDL), and Air Force Test Pilot School (AFTPS). This project utilized the NT-33A as both the flight vehicle and the ground simulator[9]. The data flights were flown at Edwards AFB and Buffalo, N.Y between July 1986 and May 1988. During this program, the piloting tasks consisted of several compensatory tracking tasks presented to the pilot on his HUD. The HUD symbology was the pilot's only visual cue since the outside visual scene was obscured from the pilot by means of a blue/amber vision restriction device. The HUD-generated tracking tasks consisted of a step-and-ramp task in pitch and roll, sum-of-sines tracking, and disturbance regulation. The aircraft configurations and corresponding task difficulties which were evaluated were representative of four generic types of aircraft - a small military transport (C-21), a fighter (F-16), a large aggressively flown aircraft (C-17), and a large transport (C-141). Varying amounts of time delay were added to the base flight control systems of the configurations to determine the degree of handling quality degradation with increasing time delay.

The companion ground simulator phase of this experiment utilized the NT-33A as a fixed-base ground simulator. The cockpit, tracking tasks, and configurations were all the same as in-flight. The only difference with regard to aircraft set-up was that aircraft motion variables were calculated by an external computer rather than the onboard sensors of the NT-33A. From a pilot's perspective, the task and procedures were the same except that on the ground there were no motion cues.

The results of this study produced a tentative design criteria for an acceptable level of additional time delay produced by mechanization of an aircraft in a ground simulator. A design goal should be to not add more than 50 milliseconds of delay between the pilot's force command and his pitch rate and roll rate motion cues. This figure assumes that the simulated aircraft and flight control system contains the 100 ms or less delay specified by the aircraft flying qualities Mil Spec, MIL-F-8785C, so that the total simulator delay from stick input to response is less than 150 milliseconds. The experiment did show differences in the effects of added delay when encountered in a demanding task environment compared to a more benign environment (Figure 4). This implies that ground simulators designed for training aircrew for unaggressive piloting tasks such as instrument flying can tolerate somewhat more added delay. The above design objective for time delay was based on the handling quality evaluations performed in the full cockpit motion NT-33A portion of the experiment. Ratings received for the fixed-based simulator evaluations had poor correlation with the NT-33A in-flight evaluations for the tasks which contained aggressive maneuvering such as the F-16 (Figure 5a). This is most probably due to the lack of cockpit acceleration cues in combination with the very austere visual references provided by the HUD. The handling qualities trends for the benign aircraft and tasks such as the C-141 showed much closer agreement between the ground and flight evaluations (Figure 5b).

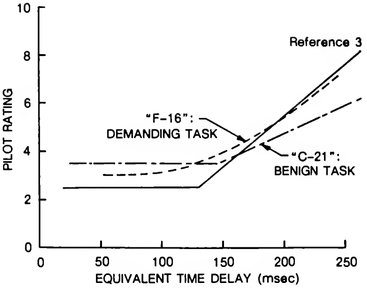


Figure 4 Influence of task demands on effects of time delay based on NT-33A in-flight evaluations

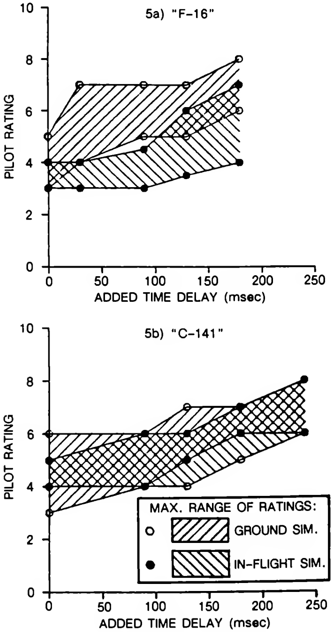


Figure 5 Comparison of ground-based and in-flight simulation pilot ratings for a fighter (5a) and transport (5b)

Simulator Time Delays

In order to know the handling qualities limitations of a flight simulator, end-to-end measurements must be made to determine the time delays present in the simulator and then compare the total response delay to the best known source of data for the response of the simulated vehicle. These responses must include all dynamic elements in the simulation system from the pilot's stick force input to the resultant aircraft motion. Since most of the research work in time delay has been done by documenting the effects of delays by measuring the angular rates of aircraft motion, an ideal situation would be to measure the pitch rate and roll rate responses of the simulator. This requires attaching a rate gyro package to the simulator cab during the simulator calibration or validation process. Time history overlays can then be generated comparing the motion response of the cab to that of the desired response, similar to what is done during the in-flight simulation validations (Figure 6). Documentation of visual delay is somewhat more complicated during ground simulator checks. Ideally, a light sensor would be placed in the cab to measure position changes of the visual scene in both pitch and roll due to a sinusoidal stick force input of various frequencies. The resulting frequency response of the display is then used to generate a pitch attitude or roll angle equivalent system transfer function including an equivalent time delay value for the simulation system. Care must be taken to account for the dynamics of the light sensor during this calibration effort[10]. Time response of the visual scene due to a step change in stick force (as opposed to frequency response) should not be used to obtain effective time delay values. The criteria established from previous flight studies have used added delays measured from pitch rate and roll rate. Delays derived from the time histories of pitch or roll attitude are significantly longer than those obtained from the rate responses merely due to the different inherent response characteristics of rate and position measurements (Figure 7).

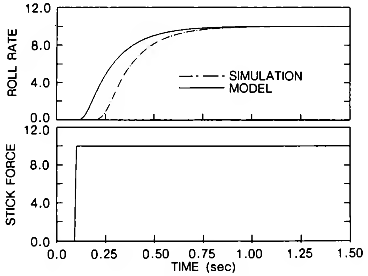


Figure 6 Example time history overlay for model vs. simulator time delay check

It is only through careful documentation of the existing time delays that an accurate assessment of the simulation's handling qualities can be made. The significance of added delay can be surmised based on a growing base of research data. For example, as described in the previous section, if the additional delay added by the simulation is less than 50 milliseconds, then the handling qualities of a high gain task is probably not significantly affected. If the simulation is found to contain significantly more delay than the simulated vehicle, the simulation should be compensated to reduce the delay. Given that a ground simulator contains more time delay than the aircraft it simulates, how can the simulation be compensated to reduce the additional delays? There are several well known ways to modify the simulation computations to reduce throughput delay[2]. In addition to changes to the simulator algorithms, judicious modification to the simulated aircraft model is often possible to reduce the delay inherent in the model and thus allow for some of the additional delay produced by the simulation. This technique has sometimes been used during in-flight simulation programs. For example, if the aircraft model contains a relatively slow control actuator, such as 15 r/s, the model of this actuator can be arbitrarily quickened to maintain the overall delay of the simulation equal to that of the aircraft. If the actuator above (assuming it is second-order) is quickened to 25 r/s the overall effective delay of the simulation is reduced by about 37 milliseconds, using the $2\zeta/\omega$ approximation for equivalent delay. Thus the motion and visual generation can add 37 milliseconds of delay without changing the total delay of the complete simulation. Similarly, some control system filters can be dropped from the simulation to reduce the overall delay. Modifications to the data base for the aircraft model must be done carefully, however, in order not to change the fundamental characteristics of the simulated vehicle. Particular care must be exercised with regard to nonlinearities since these modifications are valid only in linear systems theory.

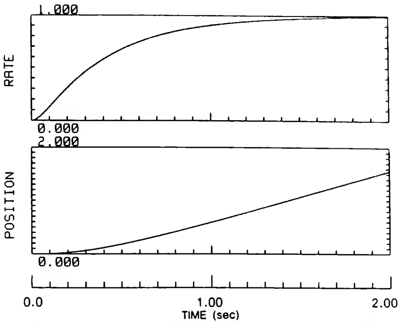


Figure 7 Second-order rate and corresponding position responses showing different time delay values

NT-33A Feel System Study

An NT-33A research program was recently conducted for the NASA Ames Research Center Dryden Flight Research Facility.[11] One of the issues addressed by this study was to determine the effect on lateral handling qualities due to variations in natural frequency of a second-order feel system. Both up-and-away and landing evaluations were conducted with an 8, 13, and a 26 rad/sec centerstick. A companion study using a sidestick was conducted by the Air Force Test Pilot School. The results of the study indicated that the lowest frequency stick (8 rad/sec) consistently produced a noticeable degradation in handling qualities for nearly all the evaluation pilots. In many cases, the pilots were not aware that the problem with the aircraft was due to the stick, but complained that the aircraft response felt nonlinear or "springy". The medium frequency stick (13 rad/sec) did not consistently degrade the handling qualities of the various configurations tested (compared to the high frequency stick) and degradation was noted less frequently during landing evaluations than during up-and-away aggressive tracking. Handling quality degradation due to low frequency feel systems occurred irrespective of whether the flight control system was a force or position command system. When low frequency flight control system filters (such as an 8 rad/sec filter) were substituted into an aircraft configuration in lieu of a low frequency feel system, the pilots also noted handling quality degradation, however the character of their complaints were completely different. This showed that the control stick is a unique element in the pilot's command path. The stick characteristics of an aircraft must be matched accurately by a simulator, and equivalent dynamic elements, which are not in the pilot's hand, can not be substituted.

The feel system damping ratio was set at .7 for all the configurations tested in the NASA study. Higher damping ratios make the stick appear to be a lower frequency to the pilots and consequently could degrade handling qualities. Feel systems with low damping ratios have not been systematically evaluated in flight, however several aircraft development programs have utilized stick damping ratios less than .2. The pilot opinions of these feel systems are that it is easy to make inadvertent, abrupt stick inputs; the pilots cannot let go of the sticks; and handling qualities are poor in turbulence.

Simulator Stick Characteristics

As with time delays the characteristics of the pilot's stick can have a significant effect on the handling qualities of a simulated vehicle. The stick force per deflection, breakout, and friction are static characteristics that can be easily measured in the ground simulator and calibrated to the desired characteristics. For these measurements a force is applied to the artificial feel system and the resulting stick displacement signal is measured. These two variables can then be crossplotted to document the characteristics as shown on Figure 8. Care must be taken to record the applied force and resulting motion at the same reference position on the stick as was done for the source data of the simulated aircraft. Small errors in reference point can result in large percentage errors when the feel system has a short pivot length such as for a sidestick.

As shown by the NASA NT-33A experiment the dynamic characteristics of an artificial feel system are equally important for an accurate simulation. A sluggish, low frequency stick can mask the properties of a flight control system and become the dominant factor in determining the vehicle's handling qualities. It is pointless to try to optimize flight control system gains during an aircraft development program if the stick hides the effects of control system gain changes from the simulator pilots. In order to provide a precise simulation, the natural frequency and damping ratio of the stick should be matched to that of the actual aircraft. If a match is not possible, a preliminary conclusion from the NT-33A feel system study is that stick frequency should be accurately matched by a simulator at least up to 13 rad/sec for landing simulations and somewhat higher if abrupt up-and-away piloting tasks are contemplated.

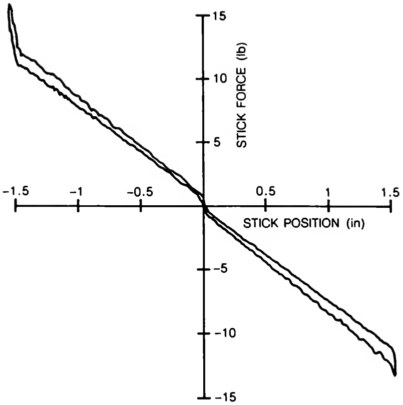


Figure 8 Example calibration plot of NT-33A stick force vs. position showing breakout, friction, deflection gradient, and position limit

Summary

In summary, several recent NT-33A programs have been used to help define simulation fidelity.

- The NASA Ames NT-33/VMS comparison study pointed out that an aircraft configuration with known poor handling qualities will not necessarily be evaluated as such in a ground simulator, even in such a capable simulator as the VMS. Well defined, aggressive pilot tasks are necessary for either ground or in-flight handling qualities evaluation projects to enable pilots to distinguish handling quality deficiencies. The presence of time delay either in an aircraft or simulator tends to cause pilots to change their control strategy to a low gain technique. In ground simulations, where there tends to be added delays due to implementation of the simulation, it is most important to enforce accurate pilot performance and to design tasks which do not allow the pilot to go open loop.

- The HRL project, which used the NT-33A for both the ground and in-flight evaluations, produced a tentative ground simulator design goal of adding no more than 50 msec of delay to the simulated model. Significant differences in handling qualities were observed between the flight evaluations and the fixed based simulator evaluations for the aggressively flown aircraft/task configurations. Less differences were noted for the benign task, lower frequency aircraft/task combinations. In order to meet the goal of adding very little delay to a simulation, some changes to the simulated model are suggested to compensate for the known minimum delays added by motion and display mechanization.

- As determined by the NASA NT-33A study of feel system dynamics, a pilot begins to be aware of handling quality effects of low frequency sticks at feel system frequencies near 13 rad/sec. Accordingly, a simulator needs to match not only the static characteristics of an aircraft's stick, but also must have a natural frequency equal to that of the aircraft if the actual stick is less than 13 rad/sec; or greater than or equal to the model for frequencies above 13 rad/sec.

Acknowledgments

The NT-33A projects discussed in this paper were performed under Air Force Contract No. F33615-83-C-3603. Mr. Steve Markman, AFWAL/FIGD, was the Air Force program manager.

References

1. Hall, Warren G. and Ron W. Huber, "System Description and Performance Data for the USAF/CAL Variable Stability T-33 Aircraft," AFFDL-TR-70-71, August 1970.

2. Gum, Don R. and Edward A. Martin, "The Flight Simulator Time Delay Problem," AIAA Paper 87-2369-CP, AIAA Simulation Technologies Conference, Monterey, Ca.,17-19 August 1987.

3. Smith, R. E. and R. E. Bailey, "Effect of Control System Delays on Fighter Flying Qualities," presented at AGARD Flight Mechanics Panel Symposium on "Criteria for Handling Qualities of Military Aircraft," Fort Worth, Tx.,April 1982.

4. Berry, D. T. et al, "A Summary of an In-flight Evaluation of Control System Pure Time Delays During Landing Using the F-8 DFBW Airplane," AIAA Paper No. 80-1626, August 1980.

5. Smith, R. E. and S. K. Sarrafian, "Effect of Time Delay on Flying Qualities: An Update," NASA Technical Memorandum 88264, Ames Research Center, Dryden Flight Research Facility, Edwards, Ca.,1986.

6. Hartsfield, Col H. W. Jr., "Space Shuttle Orbital Flight Testing," Society of Experimental Test Pilots, 22nd Symposium Proceedings, Technical Review, Volume 14, September 1978.

7. Crane, Francis, "Aircraft Pilot Induced Oscillation Tendency Evaluation - A Comparison of Results from In-flight and Ground Based Simulators," Informal report, NASA Ames Research Center, 1986.

8. Cooper, G. E. and R. P. Harper, Jr.,"The Use of Pilot Rating Scale in the Evaluation of Aircraft Handling Qualities," NASA-TND-5153, April 1969.

9. Bailey, R. E. et al, "Effect of Time Delay on Manual Flight Control and Flying Qualities During In-flight and Ground Based Simulation, AIAA Paper 87-2370, AIAA Flight Simulation Technologies Conference, Monterey, Ca., August 17-19, 1987.

10. Niemeyer, Glen, "Measurement of Flight Simulator Time Delays," AIAA Flight Simulation Technologies Conference, Monterey, Ca.,August 17-19, 1987.

11. Bailey, R.E. et al, "Interaction of Feel System and Flight Control System Dynamics on Lateral Flying Qualities," AIAA Paper 88-4327-CP, AIAA Flight Mechanics Conference, Minneapolis, MN, August 15-17, 1988.

VISTA/F-16: THE NEXT HIGH-PERFORMANCE IN-FLIGHT SIMULATOR

Gary K. Hellmann, David E. Frearson, Jack Barry Jr.
VISTA Advanced Development Program Office
Flight Dynamics Laboratory
Air Force Wright Aeronautical Laboratories

Abstract

A new, high-performance in-flight simulator aircraft is required by the Air Force to support aeronautical research and development over the next twenty-five to thirty years. The Variable Stability In-Flight Simulator Test Aircraft (VISTA) is a USAF Advanced Development Program to design, build, test, and field an improved high performance in-flight simulator using an F-16D as the host aircraft. The primary mission of VISTA will be in-flight simulation of the flight characteristics and pilot interfaces of new flight vehicles and advanced weapon systems.

Background

In-Flight Simulation

An in-flight simulator is an aircraft whose cockpit environment, stability, feel, and flying characteristics can be changed (to whatever extent possible) to match those of another aircraft. This is accomplished by means of a "fly-by-wire" variable stability flight control system and programmable artificial feel and display systems. As the pilot moves the controls, the aircraft responds as the simulated aircraft would. The pilot experiences the real flight motions, accelerations and handling qualities he would feel if seated in the cockpit of the simulated aircraft.

The key to the in-flight simulator's ability to simulate other aircraft is the variable stability system (VSS). The VSS must provide the capability to change the dynamic behavior of the host aircraft. There are two general schemes for matching the motions of the aircraft being simulated: response-feedback and model-following.

The principle of stability augmentation can be readily applied to alter the values of the stability derivatives of an aircraft. The terms in the equations of motion of the variable stability airplane can be adjusted to match the corresponding terms in the equations of motion of the airplane being simulated. This is the original variable stability concept, and it is known as the "response-feedback" approach. A response-feedback variable stability system can be described as a generalized stability augmentation system which has wide ranges of adjustment so that large variations in airplane response characteristics can be produced.

A response-feedback system operates by adding to or subtracting from the airplane's natural stability and control characteristics. This is accomplished by augmenting the stability and control derivatives of the host aircraft to the extent that they become equivalent to those of the model, and thus achieve equivalence between the responses to control inputs. Thus, it is necessary to accurately know the stability and control characteristics of the host aircraft at whatever flight test condition is being used. The use of this type of variable stability system implies a substantial task of identifying the characteristics of the host aircraft and those of the simulated aircraft at the configurations that are to be tested.

A different approach to variable stability uses the idea of "model-following". In this type of system, the electrical signals that come from the evaluation pilot's use of his cockpit controls are fed as inputs to a computer which has in it the equations of motion of the airplane to be simulated. The output of this computer is the set of time histories of motion variables which describe the response of the simulated airplane to the inputs applied by the pilot. The task then is automatic operation of the controls of the host airplane in such a way that its motions at the pilot station follow or duplicate the motions defined by the model outputs. In other words, the job of the flight control system is to make the airplane follow the pitch, roll and yaw motions that come from the model computer, and likewise to follow the indicated changes in speed components along the three axes, or equivalent variables such as angle of attack and angle of sideslip. Both the model-following and response-feedback concepts are depicted in Figure 1.

Current Capability

The high-performance in-flight simulator currently used by the Air Force is an NT-33A (see Figure 2). This aircraft is the oldest flying aircraft in the USAF inventory, having been delivered in 1951. It was developed into an in-flight simulator from a standard T-33 in the late 1950s. The demands of new aircraft have become more complex, and despite system improvements, the NT-33A is no longer representative of modern high-performance aircraft. The specific deficient areas are basic aircraft performance limitations, variable stability system limitations (the NT-33A uses only the response-feedback technique), and logistic supportability.

The NT-33A has simulated nearly every new fighter aircraft that has entered the Air Force inventory, plus several aircraft for the Navy, NASA, and allied nations. The NT-33A has also performed research to help develop a Military Specification for flight control and handling qualities.

In 1976, the Air Force and Navy Test Pilot Schools each incorporated stability and control instruction flights, using the NT-33A, into their curricula. These flights help new test pilots, who are already highly experienced military pilots, to understand stability and control variations during large maneuvers and high-g load factors. This type of realistic and cost effective training can only be achieved by the use of in-flight simulation.

The NT-33A remains active, flying about 350 hours per year for research and development programs and two sessions per year at each of the Test Pilot Schools. It is expected to maintain this schedule until it is replaced.

The other major in-flight simulator used by the Air Force today is the Total In-Flight Simulator (TIFS). The TIFS airplane is a C-131 modified to obtain independent control of the six degrees of freedom of motion of the vehicle. In addition to the usual controls, TIFS has been configured to provide for variable thrust, variable side forces by the addition of movable surfaces mounted on the wings, and variable lift forces through the use of direct lift flaps. TIFS employs the model-following technique through a complex hybrid computer architecture.

VISTA Program Conception

Given the current situation with the NT-33A, it was decided that a new high-performance in-flight simulator aircraft is required by the Air Force to support aeronautical research and development over the next twenty-five to thirty year period. Towards that goal, in 1982, a contract was awarded to Calspan to define the requirements for the next generation high-performance in-flight simulator. The program became known as the Variable Stability In-Flight Simulator Test Aircraft (VISTA) program. Final results of this contract are documented in AFWAL-TR-3021.

The study identified the following requirements: (1) VISTA must be a two-seat fighter-type aircraft, the front cockpit to be the evaluation cockpit with variable feel controllers and programmable displays and the rear cockpit to be the pilot-in-command (safety pilot) station; (2) the VSS must be capable of all-attitude model-following and response-feedback; (3) VISTA must be able to control the six-degree-of-freedom forces and moments to satisfy mission and simulation fidelity requirements; (4) VISTA must be inexpensive to maintain and operate, particularly in the Test Pilot School environment; and (5) VISTA must be supportable for the next twenty-five years.

The VISTA definition study considered several current, first-line, domestic, high performance jet aircraft to determine their suitability to perform the VISTA mission. While other aircraft met the requirements to various degrees, the F-16D was determined to be the most suitable. The selection was supported by HQ TAC and HQ USAF by the assignment of a new F-16D production aircraft to the VISTA program.

Through agreement with the F-16 System Program Office, the VISTA/F-16 will be modified during production by General Dynamics, Ft Worth Division, through a Contract Change Proposal to the F-16D Air Vehicle Contract. Production options and changes necessary to the VISTA configuration will be incorporated during procurement, fabrication, and assembly. The modification will provide the airframe/wing/landing gear structural integrity and internal volume provisions identified by the VISTA definition study. Specific modifications include the substitution of a type/version 4K (Peace Marble II modification) airframe for the 5D airframe, RAPPORT III canister, spin chute provisions, a production digital flight control system and deletion of the 20mm gun system and selected avionics. In addition, the aircraft will be delivered with an F110-GE-100primary propulsion engine with a corresponding large inlet module.

The VISTA development approach has been dictated by the available program dollars. A three phased modification program has been structured that would provide a useable capability at the conclusion of Phase I and growth provisions for Phases II and III enhancements. Core VISTA, as the Phase I product is called, will have a limited, subsonic five degree-of-freedom capability, hydraulic system upgrades, a response feedback VSS, and extensive cockpit modifications. Phase II would add a model-following capability, extend the simulation envelope to supersonic, and include the development of a ground console for: (1) VSS and graphics system software development and verification, (2) real-time simulation, (3) data analysis, and (4) system maintenance. Structural modifications to the wing and aft fuselage required to implement direct sideforce, yaw pointing and increased drag modulation would be accomplished as Phase III. The Core VISTA configuration is depicted in Figure 3.

Core VISTA Program

VISTA Program Overview

The Core VISTA program contract was awarded to the contractor team of General Dynamics, Ft Worth Division, and Calspan Corp in June 1988. General Dynamics is the prime integrating contractor. The contract is written such that, during the length of this basic contract, the Government may exercise options to incorporate some of the features that have been deferred. These options include: improved integrated servoactuators, programmable displays in the evaluation cockpit, variable-feel sidestick and rudder pedals, closed-loop control of the speedbrakes, and raised multi-function displays in the evaluation cockpit.

The Core VISTA development program is a 36 month effort planned to be conducted in five tasks: (1) Preliminary Design, (2) Detailed Design, (3) Hardware Fabrication and Assembly, (4) Ground Tests, and (5) Flight Tests. Major program milestones are as follows:

S/W System Design Review:	3 MAC
Preliminary Design Review:	7 MAC
Critical Design Review:	11 MAC
Test Requirements Review:	17 MAC
Aircraft Roll-Out:	23 MAC
Flight Readiness Review:	25 MAC
First Flight:	27 MAC
Physical Configuration Audit/ Functional Configuration Audit/ Formal Qualification Review:	35 MAC
Final Report:	36 MAC

MAC = Months After Contract

Core VISTA Requirements

VISTA technical requirements were specified for the operating requirements, aircraft performance, cockpit modifications, the VSS simulation capability and computer system, flight control system, and aircraft support systems. Minimum acceptable (mandatory) requirements were specified whenever possible. Design goals (desired capabilities which would significantly impact VISTA mission performance) were specified when minimum acceptable capabilities could not be identified or when a significant improvement would result.

The operating requirements for VISTA are summarized in Table 1. The host aircraft requirements shown in Table 2, reflect both the limited five-degree-of-freedom capability and the reduced flight test envelope of the Core VISTA program.

The Core VISTA simulation control laws will be designed to be capable of high fidelity simulation of the six degree-of-freedom dynamic responses of future flight vehicles including high performance fighter aircraft in all-attitude maneuvers. The simulation system will be capable of varying the simulation flight characteristics, as shown in Table 3.

VISTA software is required to be implemented in real time Ada; exceptions to this will be made only to meet real time software execution requirements, modifications to existing host aircraft systems (such as the flight control system), and the capability to directly use customer supplied software written in Fortran.

The data recording systems were specified to be compatable with the AFTPS, Navy TPS, and AF Flight Test Center facilities. The on-board recorder will be capable of continously recording up to 400 digital and analog parameters at sample rates up to 200 samples per second for the duration of the flight. The instrumentationsystem is based on the Flight Test Center's Airborne Test InstrumentationSystem (ATIS). In addition, VISTA will have the capability to record pilot comments and evaluation cockpit displays.

Design Approach

There are three areas of the design approach that are addressed in some detail below: the operational safety concept, the VSS/host system architecture, and the ground/flight test program.

Operational Safety System

The VISTA operational safety system is based on the System-Wide Integrity Management concept developed and demonstrated on the AFTI/F-16 program and the special safety concepts employed on the current in-flight simulators. The VISTA safety system concept has the following key elements (see Figure 4).

1. The safety pilot, through his motion cues and displayed information, provides the intelligence for manual disengage in critical non-failure conditions.

2. A VISTA Integrity Management (VIM) function in the quad-redundant digital flight control system detects failures in the VSS elements and violations of safety trip limits. The function automatically transfers control to the safety pilot.

3. Equipment and system tests that detect latent failures prior to flight.

4. Redundant circuitry for manual disengagement of the simulation by the safety pilot or the evaluation pilot.

5. An emergency mode that permits the evaluation pilot to control the F-16 through the primary control laws of the digital flight control system.

6. A digital backup unit control system in the digital flight control system for the safety pilot.

VSS/Host System Architecture

Simulation and host systems interface requirements were specified in the Core VISTA Statement of Work. The various aircraft systems, such as flight control, avionics, navigation, hydraulic, mechanical, and cooling systems, should be interfaced with the simulation system. All interfaces should have minimum impact on the software, packaging and structure, yet be as simple as possible and maintain the highest level of system integrity All of the components of the simulation system should be interfaced to each other, to the maximum extent possible, by existing 1553B Mux buses. In addition, any interface with the host flight control system should have no impact on the redundancy management of the basic flight control system. The General Dynamics/Calspan proposed architecture meets these requirements.

The proposed VSS/host system architecture is shown in Figure 5. The architecture includes four MIL-STD-1553B MUX communication networks to link the VSS elements with the display elements, avionics, and digital flight control computer of the F-16 thereby providing maximum flexibility for optimum exchange of data.

The proposed VSS computers consist of several MIL-SPEC airborne computers and an I/O chasis. Each computer will have a 32-bit floating point architecture, a minimum cache memory of 128,000 bytes, and a minimum of 4 megabytes of RAM expandable to 32 megabytes. The VSS variable-feel controllers for the evaluation cockpit will be designed and fabricated in a similar manner to the NT-33A equipment. Provisions for space in the aft cockpit and digital flight control system software capability to support future installation of the tactile cue controllers will be retained.

The production F-16 digital flight control system design will be modified to incorporate the functions required to safely interface with the VSS computers and perform the VIM and engage and disengage functions. The digital flight control computer MIL-STD-1553B AMUX interface will interface to the VSS computers through software programming changes to the computer and the operational flight programs. This will minimize hardware changes to the production flight control computer.

Ground/Flight Test Program

A highly integrated test approach will be exercised. A System Test Plan covering all aspects of the ground and flight test effort will be prepared.

Software verification and validation (V&V) will be conducted on host avionic and flight control system software and the VSS software to ensure that the operational flight programs (OFPs) and new or modified hardware perform correctly. V&V testing during the host system development will confirm the flightworthiness of the digital flight control system OFP, establish that the core avionic software supports operational requirements, and verify proper operation of the integrated systems of the host aircraft. V&V testing of the VSS software includes stand-alone V&V and integrated system testing of the total VISTA system, including Failure Modes Evaluation Tests of the VSS. The VSS elements will be incorporated with the integrated host aircraft systems, and testing will focus on VIM and the VSS interfaces with the digital flight control computer and avionics. Software testing will culminate with on-aircraft ground simulation testing of the VSS.

Pilot-in-the-loop simulations (as well as non-piloted simulation) will be used initially for development, trade studies, and evaluation in areas such as handling qualities, pilot-vehicle interface, feel system evaluations, and control law development. Later in the program, actual flight control hardware will be integrated into the simulation to aid in V&V testing as well as non-piloted frequency-response tests. Finally, the VSS hardware will be integrated into the ground-based simulation to verify proper operation of the integrated VISTA system. Formal V&V and Failure Modes Evaluation Tests will be performed on the totally integrated system.

Subsystem qualification tests will be performed to verify the suitability of the new or modified items for production and use in the VISTA/F-16 aircraft. Qualification of subsystem components will be made either by demonstration of similarity to existing components or, when necessary, by testing.

Aircraft ground tests will be performed on the VISTA/F-16 to verify flightworthiness of the aircraft and its subsystems. These tests and functional checks will be performed to establish reliable operation of the aircraft to ensure crew safety. The following ground tests will be completed: (1) mass properties, (2) EMI/EMC, (3) ground vibration, (4) structural coupling, (5) flight control system, (6) static functional tests of the instrumentation and VSS systems, and (7) preflight and taxi tests.

Aircraft flight tests will be performed at both the contractor's facility at Ft Worth and EAFB. Flight testing at Ft Worth will consist of a functional flight test program comprising 12 flights. Functional checks will be made of standard F-16 systems, modified F-16 systems, the VSS system, and the instrumentation system. Following ferry to EAFB, a three-part flight test program will be accomplished. Host envelope expansion, VSS development and validation, and VSS demonstration will be flown consisting of an estimated 8, 18, and 3 flights, respectively. Flight tests will be conducted around two flight test islands: (1) low altitude, low speed, and (2) medium altitude, medium speed.

Contract Deliverables

The VISTA contract deliverables include contractor data items, VISTA unique support equipment and manuals, computer software and manuals for both the host flight control and avionic systems and the VSS system, and the VISTA aircraft itself. Following completion of the flight test program, the aircraft will be transferred to the Government through formal Aerospace Vehicle Transfer procedures. VISTA is scheduled to become operational in the summer of 1991.

Summary

The goal of the VISTA program is straight forward: develop a new high performance in-flight simulator that provides simulation capabilities and a performance envelope that satisfy current requirements, that is logistically supportable over the next twenty-five years, and that has signifcant growth potential. The Core VISTA configuration, being based on the F-16D, will be the high performance, logistically supportable in-flight simulator required to replace the NT-33A. With the eventual completion of the growth phases, VISTA will far surpass any in-flight simulator flying today. VISTA will become a national research tool, available for in-flight simulations, flight control research, and system integration research. VISTA will also continue to support the training mission requirements of the Air Force and Navy Test Pilot Schools.

VARIABLE STABILITY IMPLEMENTATION OPTIONS

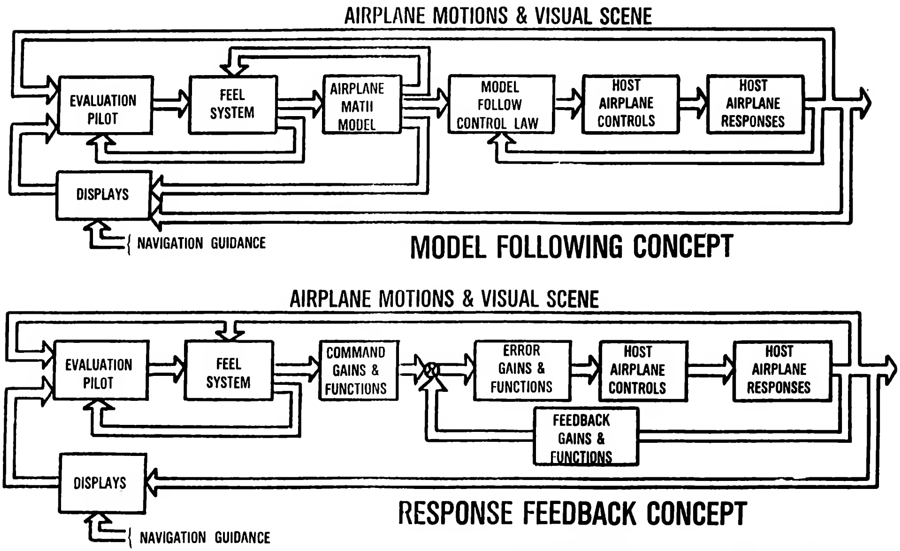


Figure 1

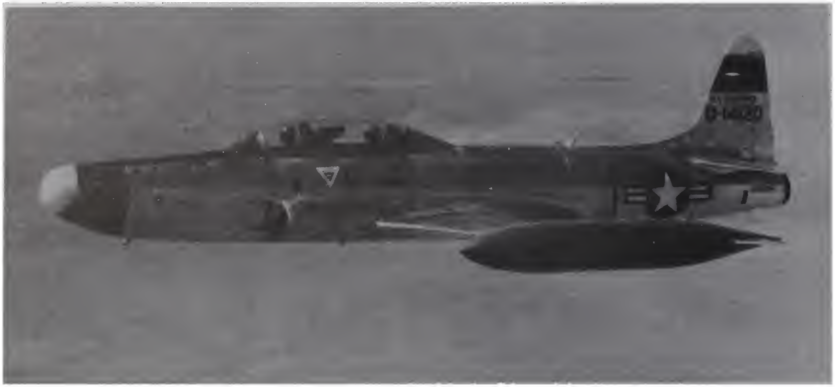


Figure 2 NT-33A

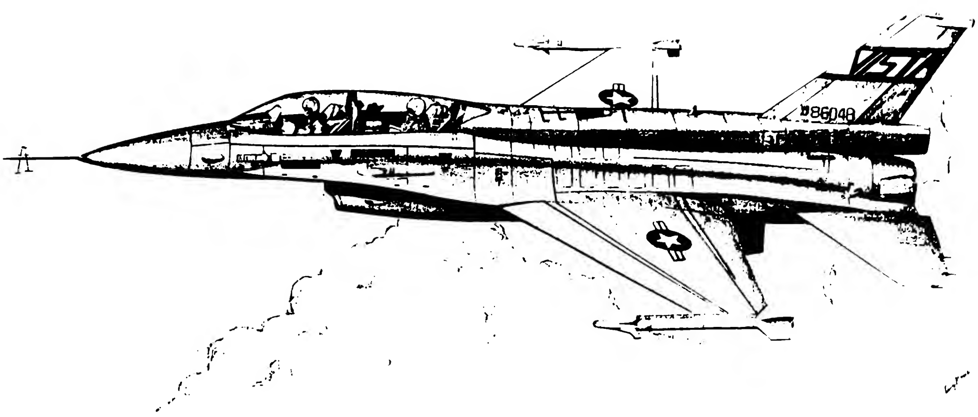


Figure 3 VISTA/F-16

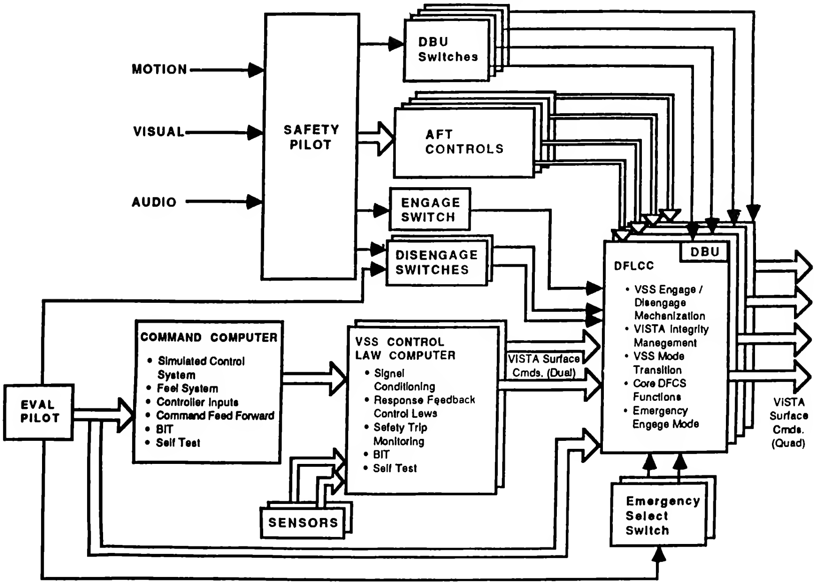


Figure 4 **VISTA Operational Safety Concept**

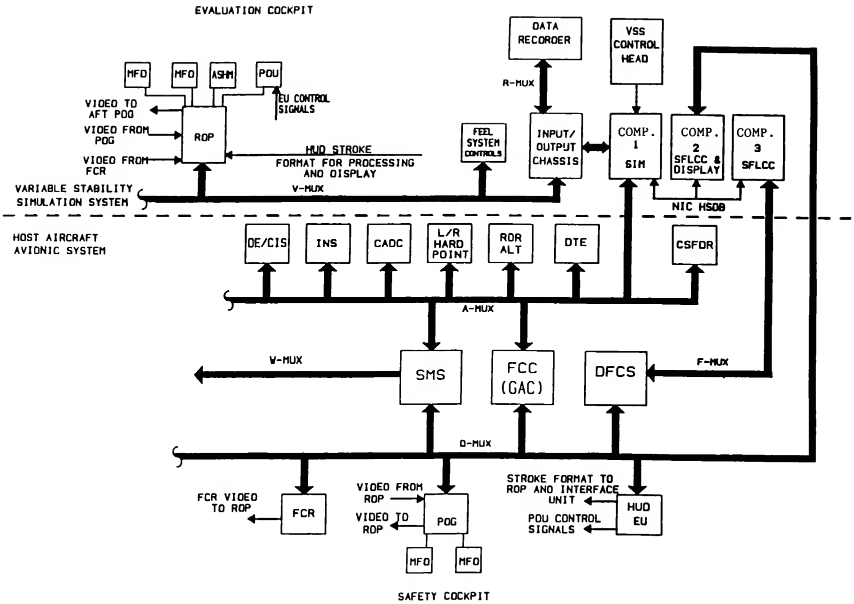


Figure 5 Core VISTA/F-16 System Architecture

Table 1 VISTA OPERATING REQUIREMENTS

Service Life (Goal)	20 Years
Flight Hours/Year (Goal)	500
Operating Costs (Goal)	$5500/Flt Hr
Ferry Range (with external tanks if required)	1200 Nm
Mission Flight Time	1.5 Hrs
Sortie Rate - TPS - Sim	3/Day 2/Day

Table 2 VISTA HOST AIRCRAFT REQUIREMENTS
(MINIMUM SIMULATION ENVELOPE)

Category	Design Goal	Proposal
Altitude (ft)	0-60000	0-50000
Min Mach Number	≤ 0.15	0.2
Max Mach Number	≥ 2.0	0.9
Max Dyn Pres (psf)	2200	680
Load Factor	-3/9	-2.4/7.33
Angle of Attack (deg)	40	16
Sideslip (deg)	10	TBD
Landing Speed (kts)	100-175	130-TBD
Sink Rate at Max Gross Weight (fps)	16	10
Roll Accel (rad/sec^2)	6 in 0.3 sec	6 in 0.5 sec
Pitch Accel (rad/sec^2)	1 in 0.2 sec	1 in 0.26 sec
Yaw Accel (rad/sec^2)	0.5 in 0.3 sec	.44 in 0.27 sec
Direct Lift (g's)	± 1 in 0.2 sec	TBD

TBD = To Be Determined

Table 3 SIMULATION FLIGHT CHARACTERISITCS

LONGITUDINAL

- SHORT PERIOD FREQUENCY AND DAMPING
- PHUGOID
- N_Z/α
- STICK FORCE PER g

LATERAL

- DUTCH ROLL FREQUENCY AND DAMPING
- ROLL MODE
- SPIRAL MODE
- ϕ/β

COMMAND FEED FORWARDS

- GAIN VARIATIONS
- LEAD/LAG FILTER
- TIME DELAY

SIMULATOR EVALUATION OF TAKEOFF PERFORMANCE MONITORING SYSTEM DISPLAYS

88-4611-CP

David B. Middleton*
NASA Langley Research Center
Hampton, Virginia 23665

Raghavachari Srivatsan**
Vigyan Research Associates, Inc.
Hampton, Virginia 23665

Lee H. Person, Jr.***
NASA Langley Research Center
Hampton, Virginia 23665

Abstract

A Takeoff Performance Monitoring System (TOPMS) to provide the pilot with graphic and numeric information pertinent to his decision to continue or abort a takeoff is evaluated. Head-down and head-up TOPMS displays were implemented on electronic screens in the Transport Systems Research Vehicle (TSRV) Boeing 737 simulator at the Langley Research Center and rated by 16 experienced NASA, Air Force, airline, and industry pilots. The head-down TOPMS display comprises a runway graphic overlaid with symbolic status and advisory information including: (1) current position and airspeed; (2) predicted locations for reaching decision speed (V_1) and rotation speed (V_R); (3) groundroll limit for reaching V_R; (4) predicted stop point for an aborted takeoff from current conditions; (5) engine-status flags and engine-pressure-ratio (EPR) bars; and (6) an overall situation advisory flag which recommends continuation or rejection of the takeoff. The simpler head-up display conveys most of this same information and relates it to the visual scene. The combined displays were well received by the pilots, who judged them easy to monitor and comprehend. In particular, the head-up display was monitored with little effort and did not obstruct or distract from the runway scene.

Introduction

Present day flight management systems generally do not provide any aids for the takeoff flight phase. Yet, statistics compiled over the years indicate that accidents in the takeoff phase account for about 12 percent of all aircraft related accidents. In recent years the accident rate in the takeoff phase has remained constant, whereas it decreased in all other flight phases.[1] - Further, most takeoff-related accidents are attributable to some form of performance degradation and a large percentage of them could have been avoided had there been a simple, yet comprehensive way to monitor the progress of the airplane's takeoff roll.

Several single-point speed checks have been proposed[2], as well as some that deal with elapsed time to reach a point on the runway[1]. Also, a multiparameter aircraft performance margin indicator[3] that continuously determines the ability of the airplane to achieve rotation speed and to brake-to-a-stop within pertinent runway constraints has been conceived. It does not, however, directly indicate where on the runway the airplane will reach V_R or where the stop point will be, but it does show the pilot how near he is to losing either his takeoff or his abort option (based on using maximum thrust for the takeoff).

An algorithm for a Takeoff Performance Monitoring System (TOPMS) has been formulated and verified[4,5] by Srivatsan under a cooperative agreement between Langley Research Center and the University of Kansas Center for Research, Inc. Subsequently, a head-down display for the TOPMS was designed and evaluated[6] on a real-time simulator at Langley by 32 experienced airline, Air Force, NASA, and industry pilots. They were impressed with the system, suggested some changes, and recommended further testing.

This paper describes the development of head-up and head-down cockpit displays to convey symbolic status and advisory information to the pilot to aid him in his decision to continue or abort a takeoff. It also documents a pilot-in-the-loop evaluation of the displays using the NASA Langley Transport Systems Research Vehicle (TSRV) fixed-base simulator. The TSRV is a modified Boeing 737 airplane[7].

Purpose and Scope

The purpose of this study was to improve the head-down TOPMS design developed in the reference 6 study, to add a compatible head-up display (HUD), and to investigate pilot acceptance of these

---

* Aerospace Engineer, Guidance and Control Division

** Aerospace Engineer, Assigned to Langley Research Center, AIAA Member

*** Research Pilot, Low-Speed Aerodynamics Division concepts in terms of appropriateness for the task, useability, and credibility. Hence, the study did not address such issues as software validation, fault tolerance, and the effects of input errors and/or noise. However, rather extensive error and failure-mode analyses were conducted when the algorithm was being developed.[4,5] The results of that effort indicated that the TOPMS distance predictions were generally within 5 percent of the actual distances computed during simulated takeoffs and aborts. The algorithm was shown to be quite sensitive to wind errors and moderately sensitive to temperature and weight input errors. It also provided the unique capability to adjust for unrealistic friction estimates, accelerometer bias and scale-factor errors.

Description of the Takeoff Performance Monitoring System

The Algorithm

The additional runway distance needed to achieve rotational speed at any instant is a function of the airplane's speed and acceleration. A simple expression for acceleration can be written as follows:

$$a = \frac{THR - D - \mu (W - L)}{m} \qquad (1)$$

There are uncertainties associated with the onboard determination of thrust in the above equation. The friction coefficient (μ) is a function of tire and runway conditions and is thus not easy to determine. Also, the lift and drag are functions of the square of airspeed. Yet, even with such uncertainties, the TOPMS algorithm[4,5] uses acceleration to generate a good composite measure of the performance of the airplane. The algorithm is composed of two parts: pretakeoff and real-time segments.

The pretakeoff segment uses detailed engine, aerodynamic, and landing-gear models in conjunction with a typical takeoff throttle movement history to generate a set of nominal airplane performance values.[5] In order to accomplish this task, the algorithm requires the inputs specified in table 1. This operation is accomplished prior to the start of the takeoff roll.

Table 1: Inputs for the Pretakeoff Segment

Airplane Center of Gravity Location
Airplane Gross Weight
Wing Flap Setting
Runway Direction
Pressure Altitude
Ambient Temperature
Wind Speed and Direction
Runway Rolling Friction Coefficient

The pretakeoff segment computes (1) the runway distance (s_o) required to attain decision speed (V_1), (2) the runway distance (s_3) required to bring the airplane to a complete stop from V_1, (3) the runway distance (s_1) required to reach rotation speed (V_R) from V_1 with one engine failed, and (4) the ground-air distance (s_2) required to attain a specified height (35 feet used in this study) at the departure end of the runway from the V_R point after experiencing an engine failure at V_1. These distances are shown in Figure 1 for the case where s_3 is less than (s_1 + s_2). [s_3 can be greater when, for example, the runway is icy.] The initial groundroll distance from the threshhold to the point where the engine failure occurs plus the greater of (s_1+ s_2) or s_3 constitutes an important metric called Balanced Field Length (BFL) or a reference minimum runway length required for the particular airplane under the existing conditions. A groundroll-limit distance to reach V_R is then computed by subtracting s_2 from the total runway length. The above-mentioned distances are somewhat conservative in that no reverse thrust is assumed for stopping and approximately 1.5 seconds is allowed for recognizing the advisory flag and initiating the abort. If the decision is not to abort, the throttles are assumed to remain in the same position as when the decision to continue was made (i.e., the algorithm assumes that the throttles are not advanced to obtain maximum available thrust on the remaining engine(s)).

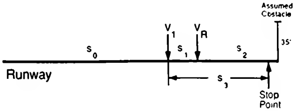


Figure 1: Components of Balanced Field Length

Table 2: Measured Inputs to the Real-Time Segment

Left & Right Engine Pressure Ratios
Left & Right Throttle Positions
Airplane Calibrated Airspeed
Airplane Accelerations
Airplane Groundspeed
Flap Setting

After the pretakeoff computations are complete, the pilot enters the length and direction of the assigned runway and how far from the threshold the takeoff roll will begin (i.e., the "runway offset"). The algorithm generates the set of nominal performance values for the current takeoff based on the estimated runway rolling friction coefficient that was entered for the pretakeoff calcuations. During the takeoff roll, the algorithm accepts the measured inputs listed in table 2, and continually calculates the airplane's present position on the runway, the runway distance needed to achieve rotation speed, and the runway distance needed to bring the airplane to a complete stop. After allowing the engine dynamics due to throttle movement to stabilize, the runway rolling-friction coefficient and the nominal performance values are recomputed. This is a unique computational feature that can be performed several times (e.g., if runway were partly dry and partly slushy); however, in this study the recalculation was only performed once. The real-time segment also monitors the "health" of the engines.

Display Format and Symbology

Figure 2 shows the location of the TOPMS head-down display in the TSRV simulator cockpit (screen in center of photo). This screen normally displays maps and data for navigation, but prior to and during the takeoff, the TOPMS display appears here. Then, immediately following main wheel liftoff, the TOPMS display is replaced by the regular navigation information.

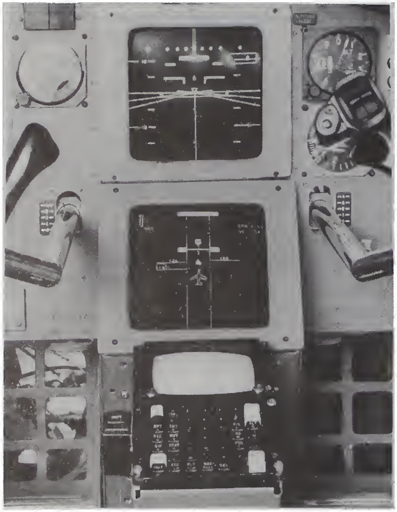


Figure 2: Location of TOPMS Head-Down Display in TSRV Simulator Cockpit

At the completion of the pretakeoff segment, the display comes up in a default mode, as shown in Figure 3. In this mode, the runway length is scaled to the calculated BFL (shown in the figure as 4834 feet). Also, at the top of the runway graphic is a rectangular colored box, which operates as a "GO/ABORT" Situation Advisory Flag (SAF). The color of this flag indicates the instantaneous advice given by the TOPMS for continuing or aborting the takeoff, as indicated in Table 3.

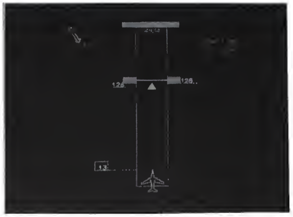


Figure 3: Head-Down TOPMS Display Showing a Balanced Field for Typical Conditions

Table 3: Colors, Sizes and Conditions for the SAF

COLOR/SIZE*	FLIGHT CONDITION
Green	(1) Takeoff is proceeding normally (2) No engines have failed; airplane can attain V_R before reaching groundroll-limit line, but cannot stop on the runway (3) One engine has failed at a speed greater than V_1; airplane can attain V_R before reaching the groundroll-limit line, but cannot stop on the runway
Flashing Amber	(4) One engine has failed at a speed greater than V_1; airplane can reach V_R before reaching the groundroll limit line, and can stop on the runway
Red	(5) One engine has failed at a speed less than V_1 (6) Both engines have failed (7) Predicted rotation point is beyond groundroll-limit line (8) Longitudinal acceleration is not within the specified error band (e.g., 15 percent) of the value predicted by the algorithm for the throttle setting being used

*In the head-down displays the SAF's are all twice as wide as the runway. In the head-up display the green SAF is the same width as the runway, the amber SAF is twice as wide, and the red SAF is three times as wide.

At the lower end of the runway graphic is an airplane symbol whose nose marks the airplane's current longitudinal position. (By choice, the airplane symbol did not move laterally in this study.) To the left of the runway symbol (opposite the nose of the airplane) is a horizontal line with a box at one end; this line further denotes the airplane's longitudinal position, and the number inside the box is Calibrated Airspeed (CAS), in knots. (The line and the box advance down the runway along with the airplane symbol.) Note in figure 3 that the nose of the airplane is about 500 feet from the starting end of the runway; this increment is called the "runway offset" and represents where the on-ramp being used intersects the runway. The takeoff roll begins here.

Further up the runway a shaded triangle is shown; its apex indicates the longitudinal position where V_R will be achieved, based on current conditions. The line to the right of the runway further denotes this position and the number (128) on the line gives V_R in knots. Similarly, the line and number to the left indicate V_1 and where it will occur.

In reality there are two triangles in figure 3--one lying on top of the other. The shaded triangle represents the instantaneous predicted point of achieving V_R and the open triangle (hidden) marks the pretakeoff prediction of this point. The open triangle is thus stationary, but the solid triangle and the V_1 and V_R lines move to indicate the updated positions.

Just above the position of the triangles is a line that stretches across the runway, with a colored box attached at either end (the boxes lie outside the runway). This line represents the groundroll limit for reaching V_R. The boxes represent engine health flags--green for engine operating normally and red for engine operating outside acceptable limits.

The arrow and the numeral at the top left of the display represent the wind direction (relative to the runway) and the wind speed in knots. The tick marks on the right edge of the runway represent 1000-foot marks starting from the takeoff end of the runway (top of picture). The recommended takeoff Engine Pressure Ratio (EPR) setting and the second segment climb speed (V_2) in knots are shown at the top right corner of the display for reference. And finally, the number under the situation advisory flag at the end of the runway gives runway length, in feet. In this case the BFL (4834 feet) for a particular set of conditions is shown; it will change to the actual length of the runway being used when that value is entered after the pretakeoff calculations are complete.

Figure 4 shows a situation with the airplane well into the takeoff run on a runway that is oriented 220° from North, hence the "22" marking on the threshold of the runway graphic. As shown in the CAS boxes to the left (and right) of the airplane symbol, the airspeed is 100 knots. [Duplication of the CAS box and line was a recommendation made by the pilots in the reference 6 study.] The apex of the unshaded triangle (stationary) represents the point at which V_R should have been achieved based on pretakeoff computations. The solid triangle, and the V_1& V_R lines have shifted upward, representing the current predictions of where decision speed and rotation speed will occur. A pair of linear bars have extended upward from the engine flags. These bars give a gross indication of the measured EPR of the left and right engines. The horizontal line just above the top of the right EPR bar shows where the top of the bar should be for the recommended EPR = 1.95; in this case, the values for both engines are low, indicating that the throttles are probably underadvanced. [An engine-performance deficiency will also cause a bar to be low, but a deficiency of the magnitude shown in figure 4 would have triggered an abort flag.] The TOPMS recommendation (green SAF at end of runway symbol) is to continue the takeoff. Note that the solid triangle indicates that rotation speed should be attained nearly 2000 feet from the groundroll-limit line.

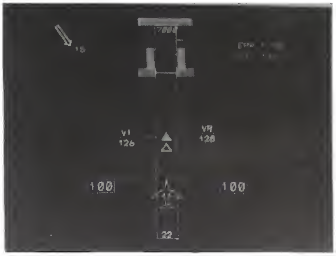


Figure 4: Head-Down TOPMS Display for a Takeoff With the Throttles Underadvanced

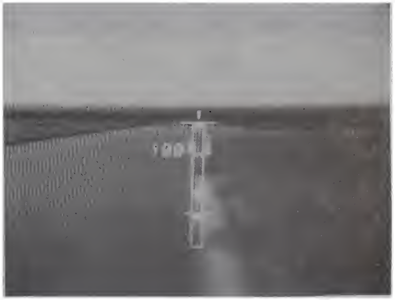


Figure 5: Head-Up TOPMS Display for Takeoff With Throttles Underadvanced

Figure 5 shows the somewhat simpler head-up display (HUD) for this same situation. The airplane symbol is represented by a solid box with a horizontal line across the front of it. The two triangles and the EPR bars are similar to those shown in the head-down display, but the CAS boxes are replaced by a large airspeed numeral (100), fixed in the center of the field of view. Other numeric information in the head-down display is not repeated in the HUD. The vertical wedge at the end of the runway is an added feature that shows the level of measured acceleration with respect to the calculated value for the throttle setting being used. In this case, the achieved acceleration matches the nominal value (to within a specified deadband) and the wedge is centered at the end of the runway. Again, the triangles have separated because of the low throttle setting--not from any engine deficiency.

An acceleration deficiency causes the wedge to move left; when it reaches the end of the scale shown (15 percent for this study), an abort flag is triggered. [For noticeability, the abort flag in the HUD was made three times as wide as the green flag, so it extends beyond the end of the acceleration scale.] This situation is illustrated in figure 6 for a recommended abort at 120 knots. An additional symbol ("X") also appears, and indicates the maximum-braking stop position; this position is updated in real time based on current position, speed, and acceleration.

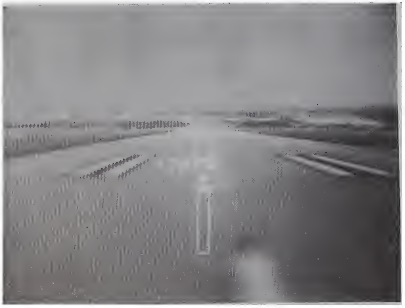


Figure 6: TOPMS Head-Up Display For a Takeoff Attempt With Unacceptable Acceleration

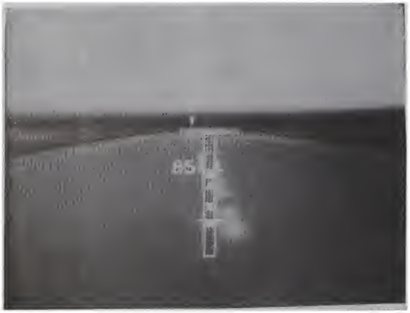


Figure 7: TOPMS Head-Up Display For a Right-Engine Failure Below Decision Speed

Figure 7 shows another abort situation--this time due to a failed engine on the right side. Airspeed is 85 knots. Again, the acceleration-deficiency wedge has shifted left and the triangles have separated significantly. But this time the EPR bar on the right side has dropped and turned red, identifying the failed engine. This same EPR pattern occurs in the head-down display; however, the width of the SAF flag remains constant for all colors.

Once an abort has been initiated, the display converts to the configuration shown in figure 8. All of the takeoff-related symbology is deleted, leaving only the airplane and the maximum-braking stop point symbols. The airspeed numeral is replaced with groundspeed and a new symbol (shaped like a football) has appeared. This new symbol indicates the stop point based on measured acceleration; in this case, less than maximum braking is being applied. The head-down display converts similarly, as shown in figure 9.

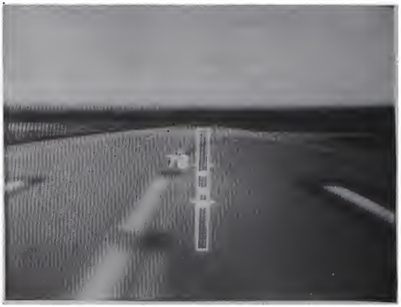


Figure 8: TOPMS Head-Up Abort Display

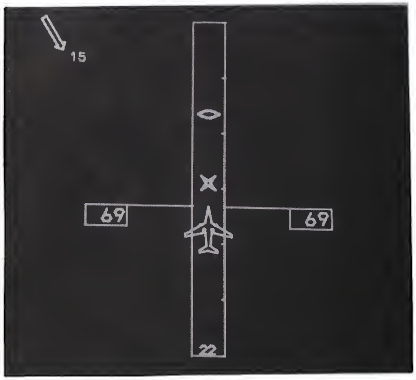


Figure 9: TOPMS Head-Down Abort Display

Description of Simulation

The simulation is accomplished with a 6-degrees-of-freedom nonlinear model of the Langley TSRV B-737 airplane, consisting of a detailed aerodynamic package, an engine model, and a landing gear model. The aerodynamic package incorporates two- and three-dimensional table lookups for aerodynamic coefficients and adjusts them for ground effects. The engine model includes detailed ram-air and temperature effects. The landing gear model provides for braking and steering. The simulation package also contains modules that provide realistic sensor-noise effects.

TSRV Simulator Cockpit

Pilot interface to this simulation model is accomplished through a fixed-base replica of the research flight deck of the TSRV (fig. 10). This simulated cockpit incorporates nearly all the features found in the aft flight deck of the actual TSRV. Each pilot has two CRT displays and a Navigation Control Display Unit (NCDU) arranged in front of him. In addition, they share a pair of engine displays (CRT's on the center panel between them). A simulated out-the-window runway scene (not shown in fig. 10) was provided for the pilot in the left seat only. This is the background scene that appears in figures 5-8.

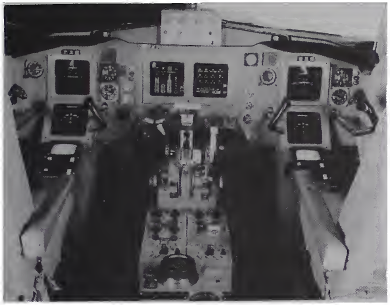


Figure 10: Cockpit of Langley TSRV B-737 Simulator

The upper CRT (directly forward and just below the glare shield) is a modified attitude display, referred to as the Primary Display. The CRT below it normally serves as the Navigation Display, which displays the maps, waypoints and other data used for airborne navigation. In this study it also displayed the TOPMS information while the airplane was on the runway.

Below the Navigation Display is the NCDU. It consists of a small black and white CRT display and an alpha-numeric keypad (see fig. 2). The pilot uses this unit to enter navigational data and other information into the flight computer; it also serves as the pilot's input device for TOPMS data.

TOPMS Operation

The Takeoff Performance Monitoring System, as mentioned earlier, consists of two parts. The first part, the pretakeoff segment, is activated prior to the start of the actual takeoff roll as follows: The pilot, using the NCDU, enters the information listed in table 1 and then activates the pretakeoff computations. Once these computations are complete, the Navigation Display screen produces a default TOPMS display like the one shown in figure 3. The HUD shows a similar graphic. The pilot then enters the actual runway length and both displays are updated accordingly. The system is now ready for takeoff.

During the actual takeoff roll, the pilot-flying moves the throttle to an intermediate setting, waits for the EPR to reach an intermediate value and then moves the throttles to near the recommended takeoff setting; the other pilot makes the final adjustments. When rotation speed is reached, the pilot pulls on the modified column (see fig. 2) until the airplane's pitch attitude reaches about 20°; he then returns the column to neutral. As the wheels lift off the runway, the TOPMS display is transparently replaced by selected map displays.

Evaluation

The TOPMS is being evaluated in several phases. The algorithm was analyzed and verified in batch simulation[5] for accuracy and sensitivity to various input parameters. Then, an initial TOPMS head-down display was designed, tested, and rated[6] by over 30 experienced multi-engine pilots on the Langley TSRV B-737 real-time simulator. Based on their comments and suggestions, the displays were revised as shown by the foregoing pictures and have been evaluated on the same simulator. Subsequently, a selected or "final" configuration will be programmed on the flight computer of the actual TSRV B-737 airplane and flight tested.

The real-time simulation sessions each involved two pilots working as a crew. The evaluation-pilot population for this study is shown in table 4. Most had not met or worked together before. The subject pairs received a short writeup and a 10-minute video prebriefing on the system before proceeding to execute a program of takeoffs or aborts while monitoring the TOPMS display for approximately 2 hours (1 hour as the pilot-flying

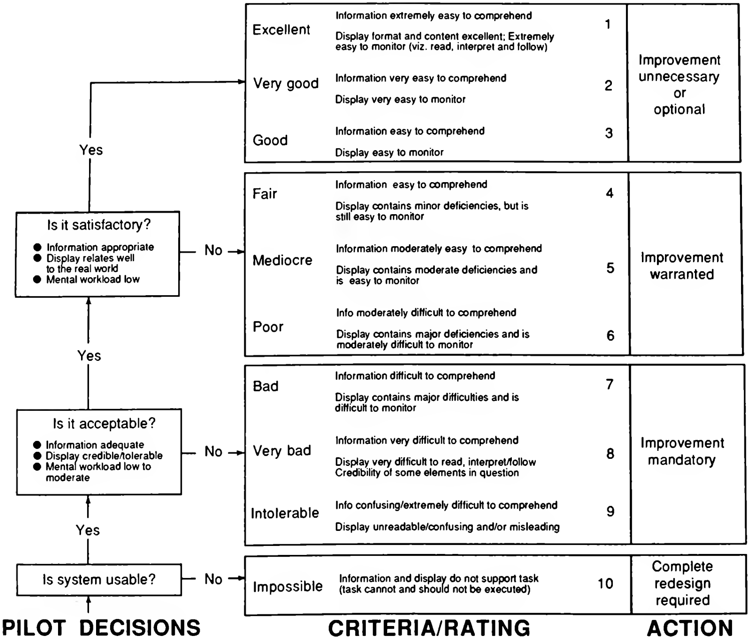
Table 4: TOPMS Evaluation Pilots

Pilot Categories	No. of Pilots
Air Force*	6
Airline**	7
- American	
- Delta	
- Piedmont	
- United	
Other***	4
Total	17

*KC-135 Tanker Pilots from Langley Air Force Base, Hampton, VA

**Pilots provided by Airline Pilots Association (ALPA) Safety Office, Herndon, VA

***Includes NASA and Retired Industry Pilots and 1 hour as the pilot-not-flying). During several practice runs at the beginning of the session, the pilots agreed on their division of duties/operating procedures (e.g., what speeds or events the pilot-not-flying would call out to the pilot-flying). The session itself consisted of approximately 20 runs for each pilot, covering a variety of conditions. In particular, the conditions included normal takeoffs, reduced-thrust takeoffs (both intentional and as an error condition), ambient temperatures from 0 to 100 degrees Fahrenheit, pressure altitudes from sea level to 5000 feet, a variety of wind conditions (both known and as an error condition), numerous runway lengths, light-to-heavy gross weights, unannounced deployment of spoilers (to create excess drag), dry and slushy runway surface conditions, and several combinations of the above. [The runs were selected to exercise, as a minimum, all of the flag conditions listed in Table 3.]



Excellent
Information extremely easy to comprehend
Display format and content excellent; Extremely easy to monitor (viz. read, interpret and follow)
1
Very good
Information very easy to comprehend
Display very easy to monitor
2
Good
Information easy to comprehend
Display easy to monitor
3
Improvement unnecessary or optional
Fair
Information easy to comprehend
Display contains minor deficiencies, but is still easy to monitor
4
Mediocre
Information moderately easy to comprehend
Display contains moderate deficiencies and is easy to monitor
5
Poor
Info moderately difficult to comprehend
Display contains major deficiencies and is moderately difficult to monitor
6
Improvement warranted
Bad
Information difficult to comprehend
Display contains major difficulties and is difficult to monitor
7
Very bad
Information very difficult to comprehend
Display very difficult to read, interpret/follow
Credibility of some elements in question
8
Intolerable
Info confusing/extremely difficult to comprehend
Display unreadable/confusing and/or misleading
9
Improvement mandatory
Impossible
Information and display do not support task (task cannot and should not be executed)
10
Complete redesign required
Yes
Is it satisfactory?
● Information appropriate
● Display relates well to the real world
● Mental workload low
No
Yes
Is it acceptable?
● Information adequate
● Display credible/tolerable
● Mental workload low to moderate
No
Yes
Is system usable?
No
PILOT DECISIONS
CRITERIA/RATING
ACTION

Figure 11: Evaluation Chart Used For Rating the TOPMS Displays

At the conclusion of the simulation session, each pilot was asked to independently evaluate the system by (1) making free comments, (2) answering specific questions, and (3) giving a separate "goodness" rating for the head-up and the head-down TOPMS displays. The rating (on a scale of 1-10) was extracted from the Figure 11 diagram, which is similar to the one used with the Cooper-Harper Scale (Ref. 8) for aircraft handling qualities. The numerical ratings are averaged for categories, but are not otherwise treated statistically because of their subjective nature.

The pilots were instructed (in writing and verbally) not to let factors such as unfamiliar controls and instrumentation or the location of the TOPMS in the simulator cockpit influence their rating of the TOPMS display per se. (The pilots were, however, encouraged to identify desirable or undesirable features of the overall simulation).

RESULTS

As previously discussed, two types of results were obtained: (1) solicited and unsolicited comments and (2) display ratings. The solicited comments were primarily answers to a set of questions (distributed to the pilots in advance) supporting the rating scale and to an undisclosed set of debriefing questions asking how such a system might be used and/or accepted by the pilots. The questions also polled the pilots' preference for particular elements or symbols in the display. As shown in Figure 11, the rating scale was based on criteria related to the appropriateness of the displays for the task and how easy/difficult it was for the pilot to extract and comprehend the data, particularly during a relatively quick scan. Other criteria included credibility, compatibility with other cockpit information and with the runway scene (simulated), and effect on mental effort when this display is integrated into existing cockpit instrumentation and/or the visual scene.

Pilot Opinion and Comments

In general, the pilots were impressed with the features of the TOPMS and would like to see this type of information available in their cockpit both in a head-down display and as a head-up overlay of the visual runway scene.

The more significant comments offered by the "pilot-flying" include:

1. The TOPMS head-up display (HUD) enhances the information obtained from the visual runway scene. It concentrates all information pertinent to the pilot's "GO/ABORT" decision in his normal field-of-view (out-the-window) during the takeoff roll.

2. The HUD does not distract from the visual scene nor does it block out any critical runway information. In fact, the relative motion between the HUD and the out-the-window scene proved to be a guidance aid for steering the airplane to the runway centerline (or parallel to it).

3. The pilot-flying tended to "look through" the HUD during most of a normal takeoff and not be strongly aware of its elements except for the airspeed numeral. His primary visual awareness was of the edges of the runway and whether he was going straight down the runway. He commented that he was subconsciously alert for changes in the EPR bars, triangle separation, appearance of the stop-point ("X"), and (particularly) SAF change.

4. The HUD did not cause the pilot-flying to rely less on the pilot-not-flying; he still preferred that the pilot-not-flying have primary responsibility for monitoring the TOPMS (on the head-down display) and orally apprising him of particular speeds and performance anomalies (such as triangle separation). After a few runs, several pilots said they relied on the large CAS numeric on the HUD and did not look back inside at the instrument panel to verify the value on their regular airspeed indicator (round-dial). They further said that they were relying on the pilot-not-flying to make this cross-check.

5. The acceleration-performance wedge received a mixed reaction: many of the pilots ignored it because it was an unfamiliar cue and it did not behave in a natural way; that is, they did not associate leftward movement across the runway with lack of acceleration in the longitudinal direction. On the other hand, several pilots quickly understood its message and appreciated the advance warning when an unacceptable performance deficiency was developing.

6. The pilots did not like the abstract airplane symbol in the HUD.

Comments offered by the pilot-not-flying include: [Eight of the evaluation pilots had also seen and evaluated the initial head-down TOPMS display during the Reference 6 study.]

7. The head-down display provided useful information even before the takeoff began (viz., all the information shown on Fig. 3). After the offset and actual runway length were entered, the display showed both pilots how much margin they had between the rotation point (triangle) and the groundroll-limit line.

8. The EPR bars were a desirable addition to the head-down display; however, they should replace, not be in addition to the engine flags. Also, they are not quite adequate for setting the thrust; the more accurate regular engine instruments should be used for this task.

9. The duplicated CAS box and line on each side of the airplane symbol was favored by some pilots and considered superfluous by others. Both groups agreed that having both would be useful if there were a greater difference between V_1 and V_R . The pilots also agreed that the CAS-line-closure on the V_1 (or the V_R) line was a very useful cue. Some pilots made the V_1 speed call based on the crossover rather than reading the CAS number.

10. The pilots-not-flying speculated that they would prefer the head-down display to be located higher on the instrument panel so they could also look out the window more easily during the takeoff roll. [The pilots-not-flying had no out-the-window scene during this simulation study.]

The pilots agreed (unanimously) that conversion from the Takeoff Display to the relatively simple Abort Display in both the head-up and head-down locations was desirable and that the transition occurred transparently. Also, the pilots said that they would like to see the Abort Display adapted to landing.

Pilot Ratings of the Displays

Sixteen of the pilots rated the TOPMS using the flow diagram/scale shown in Figure 11; the extra pilot did not "fly" the full set of runs and preferred not to give a numerical rating (although he gave comments). Separate ratings were obtained for the head-down and head-up configurations as indicated in Figure 12. The ratings in this figure are averaged for the groups shown in Table 4. In general, the ratings for the head-down display were about the same ("3" or "good") as they had been for the initial display evaluated in the Reference 6 study. The ratings for the HUD presentation ("2" or "very good") were considerably better than for the head-down display.

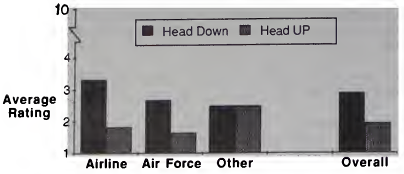


Figure 12: TOPMS Ratings by Pilot Work Group

CONCLUDING REMARKS

The displays evaluated in this study provide the first indication of how pilots might accept and use the combined head-up and head-down TOPMS information. The displays were well received by all of the evaluation pilots, who encouraged continued development. They rated the head-down display "satisfactory". In particular, they felt that in its tested form/location, it would require deliberate but low-effort monitoring. They agreed (unanimously) that the pilot-not-flying should have primary responsibility for monitoring the information and announcing events and/or advisories to the pilot-flying. The simpler head-up display (HUD) enhanced the visual information for the pilot-flying and proved also to be a steering aid. It was rated "very good". After a few runs, most of the pilots-flying began monitoring airspeed (primarily) on the HUD (in conjuction with speed calls made by the pilot-not-flying). [It is not clear if this is a good or bad feature, but as a mental task, it was declared preferrable to looking back inside the cockpit.] The pilot-flying tended to "look through" the HUD graphics and was only aware of significant or sudden changes in the symbology. The symbology did not mask out any useful cues from the visual scene. The abort displays were well-liked and no improvements were suggested.

It is recognized that substantial engineering problems related to implementation of a "working" TOPMS still exist. They include software validation and provision of improved performance data to the algorithm. In particular, determination of proper runway friction coefficients for wet and contaminated surfaces remains elusive. It is also recognized that technology to provide a constantly sharp, clear color HUD display (for all lighting conditions) needs to mature before implementation will occur. However, this study has provided a step forward by verifying desirable and quite useable takeoff/abort advisory display configurations and dynamics.

REFERENCES

[1] Minutes of the first meeting of the SAE Takeoff Performance Monitoring Ad Hoc Committee of the Aircraft Division of the Aerospace Council: May 15-16, 1984, Washington, D.C.

[2] Small, Maj. J. T.: "Feasibility of Using Longitudinal Acceleration (N_x) for Monitoring Takeoff and Stopping Performance from the Cockpit"; Proceedings of the Twenty-Seventh Symposium, Beverly Hills, CA - September 28 - October 1, 1983, Pages 143-154; (A84-16157 05-05).

[3] Cleary, Patrick J.; Kelman, Lloyd S.; and Horn, L.: "Aircraft Performance Margin Indicator". U.S Patent Number 4,638,437, January 20, 1987

[4] Srivatsan, R.; Downing, D. R.; Bryant, W. H.: "Development of a Takeoff Performance Monitoring System"; Presented at the AIAA Guidance, Navigation and Control Conference, Williamsburg, VA, August 18-20, 1986; see also NASA TM-89001.

[5] Srivatsan, R.: Design of a Takeoff Performance Monitoring System; Doctoral Dissertation, University of Kansas; June 1985. Also published as NASA CR-178255; March 1987.

[6] Middleton, David B.; and Srivatsan, R.; "Evaluation of a Takeoff Performance Monitoring System"; Paper No. 87-2256-CP, Presented at AIAA Guidance, Navigation, and Controls Conference, Monterey, CA, August 17-19, 1987.

[7] Staff of NASA Langley Research Center and Boeing Commercial Airplane Company;"TERMINAL CONFIGURED VEHICLE PROGRAM--Test Facilities Guide"; NASA SP-435; 1980.

[8] Cooper, G. E., and Harper, R. P., Jr.: "The Use of Pilot Rating in The Evaluation of Aircraft Handling Qualities"; NASA TN D-5153; April 1969.

88-4612-CP

SMART COMMAND RECOGNIZER (SCR): FOR DEVELOPMENT, TEST, AND IMPLEMENTATION OF SPEECH COMMANDS

Dr. Carol A. Simpson*
Psycho-Linguistic Research Associates, Woodside, CA 94062

John W. Bunnell
SYRE, Inc., NASA Ames Research Center, Moffett Field, CA 94035

Robert R. Krones
Informatics Division, Sterling Software, Palo Alto, CA 94303

Abstract

The Smart Command Recognizer (SCR) is a rapid prototyping system for developing, testing, and implementing speech commands in a flight simulator or test aircraft. A single unit performs all functions needed during these three phases of system development. The use of common software and speech command data structure files greatly reduces the preparation time for successive development phases. The SCR provides for speech template enrollment and testing, a training mode to teach an application command set to user pilots, and a speech command cockpit procedures training mode which provides interactive speech command training in a functional cockpit environment. Prompting and feedback are given by visual displays and a synthesized speech annunciation system. The SCR operates in a stand-alone mode for speech command development and in a host mode to provide speech command input to a host computer. As a smart peripheral to a simulation or flight host computer, it handles all interpretation of the pilot's spoken input and passes command codes to the simulation or flight computer. This reduces the execution load on the host computer and eliminates the need for duplicate programming to implement an already tested speech command system on another computer. Software resident in the host simulation computer causes experimenter-specified sequences of flight functions to be executed in response to specific commands received from the SCR. This configuration is an extremely powerful development, test, and implementation ᵗool for cockpit speech command systems.

The Problem

Many factors must be considered in the design of an efficient speech I/O system. Efficiency is particularly important for speech command systems that will be used for control of combat aircraft. From the pilot's point of view, the vocabulary and word order of speech commands must be easily remembered. The speech input system must give the pilot feedback on its status, including what it has recognized, whether it can act on what it has recognized, and whether the speech signal getting to the system is of adequate quality in terms of signal level and signal-to-noise ratio. From the perspective of the recognition device, the vocabulary items must be acoustically dissimilar, the command syntax must be unambiguous, and the number of recognition choices that result in valid speech commands must be kept to a minimum to minimize processing time for recognition template matching.

The development cycle for the vocabulary and command wording for a particular application usually involves several iterations of vocabulary selection and command wording design. Here is a typical scenario. A set of candidate functions for control by speech are selected by some sort of criteria for alleviating heavy visual and/or manual workload. An initial vocabulary is chosen and tested with a few speakers using one or more speech recognition devices. Problem words are found which are erroneously substituted for each other by the recognition devices, due to phonetic or acoustic similarities. Synonyms for these problem words are selected, the recognition tests are repeated, and it is discovered that the new words are similar to some other words in the vocabulary. Iterations of this process are unlikely to succeed in eliminating recognition substitutions.

The next stage is often to organize the vocabulary into sets of words, according to the syntax, i.e. word order, of the commands. An initial set of words, those which can start a command, is made active for recognition. When a word from that set has been recognized, then a new set of words, those which could legally follow the word already recognized, is made active. By limiting the "active recognition set" to only those words that can legally follow the word or words that have been previously recognized, the designer reduces the chances for substitution errors by the recognizer. Another approach to using syntax to improve recognition is to put all words in the vocabulary into a single active recognition set. Then a posthoc elimination of impossible candidate sequences of words may be performed according to the constraints imposed by the legal command syntax.

Assignment of words to different sets or nodes in the syntax is dictated by the need to minimize the acoustic and phonetic similarity of words within each set. Sometimes new words are added to the vocabulary to serve as "branch" words to direct the recognizer to one of several possible subsets of words in the attempt to separate acoustically similar words. The user is brought in to try the new commands at each iteration. The user is unable to remember some of the synonyms because they are not in common usage in his or her cockpit. The user also has trouble remembering the command word order or may forget some of the artificial "branch" words. As a result, he or she frequently says words that are not legal for the current active recognition set. Accordingly, the vocabulary and command wording are changed by the designer again.

Each time a vocabulary or command wording change is made, the program that controls the speech recognizer must be changed. For some speaker-dependent speech recognition devices, the templates for the vocabulary words must be reenrolled. Speech command design thus becomes a long and expensive process. Eventually, the design is completed. Then it must be implemented on the target system: a flight simulator, an aircraft, or a command and control station, in order to let the pilots use it in context. More programming is needed, much of it to implement processes that were already implemented in the test system in the lab.

The Solution

This paper describes a development tool that eliminates this duplication of programming and which makes the entire design and test cycle much easier and more efficient. In addition to the functions that sometimes are available in vendors' speech recognition development systems, the SCR provides functions that address the designer's need to iteratively test and modify vocabulary, speech templates, command syntax, and type of system feedback. Speech recognition data, collected and stored on disk, can be used to assess recognition accuracy and pilot ability to remember command wording. Vocabulary words, templates, command syntax, feedback mode, or feedback content can each be changed independently of one another and without the need for any software changes. Following the development phase, the SCR is interfaced to the host computer in the simulator or aircraft as a peripheral device.

System Overview

The SCR is a software package, written in C, that runs on an IBM PC-XT compatible microcomputer under the MS-DOS operating system. Peripheral devices for the microcomputer include a speaker-dependent, connected word recognition device, a visual feedback display, a phoneme speech synthesizer, an experimenter console, and a printer for data file printout. The SCR has two modes: a stand-alone mode for speech command development and testing, and a host mode, interfaced to a host computer. In the stand-alone mode, the visual feedback display is a color monitor for presenting text messages to the user pilot. In the host mode, it is an 80-character, one-line electroluminescent display connected to the parallel port of the SCR computer. The printer is not used for the SCR host mode.

SCR Stand-Alone Mode

The SCR in stand-alone mode has two main programs. The Template Enrollment program is used to enroll and test speech templates. The Recognition program is used to test templates in the context of the application command wording, to train pilots on the correct command wording, to test their ability to remember the command wording, and to let them practice with the command wording in a functional cockpit procedures trainer.

The SCR's Command Syntax Interpreter is one of its more powerful features. The syntax interpreter was designed to allow an experimenter a great deal of versatility in designing the command syntax for a target application. A brief description is given here.

SCR Command Syntax Interpreter

The Command Syntax Interpreter distinguishes among three levels of words. Command words are the normal words that make up commands. Data words supply some sort of data to complete a command. Meta words are words that compose commands to exercise control over the recognition process.

Command Words. Command words can be strung together within the constraints of an experimenter-defined syntax or word order, to compose commands. A command can be a single word or a phrase of words. The Command Syntax Interpreter obtains all its information about the syntax of a particular set of speech commands from a text file that is created by the experimenter with any text editor. This file, called the Set File, can be edited and reused, making the process of changing command syntax very easy. The Interpreter permits several characteristics of natural language including synonomous lexical elements and syntactic structures, context-sensitive meaning, and limited recursion. Details are giyen in a previous paper (Simpson and Krones, 1988).[4]

Data Words. Data words are unconstrained by syntax. A word functioning as a data word can appear in any position within a string of data words. A common use of data words is for supplying alphunumeric data to complete a command, such as a call sign in conjunction with a command to tune a radio to the frequency associated with the call sign of a particular individual.

Meta Words. Meta words let the pilot control the command recognition process. Current meta words are:

ENTER	terminate alphanumeric data en
DELETE	delete the previous word
DISREGARD	cancel a command
STANDBY	save the current command and s
RESUME	load a saved command

Template Enrollment

The Enrollment Program is used to enroll templates for those vocabulary words that are needed for a particular speech command syntax.

Vocabulary words can be presented to the pilot for enrollment in any order. Any subset of the vocabulary can be enrolled at a time without regard to the assignment of words to syntax nodes or recognition sets. Selection of words for enrollment and control of the enrollment process can be interactively controlled by the experimenter or can be performed automatically using a previously edited text file, called a Sequence File. The pilot is prompted on the word to say next. The word appears in text on the color monitor. The spelling of the word is taken from a previously edited text file, which is called the Vocabulary File. The Vocabulary File and the Sequence File can be modified independently and reused.

The Enrollment Program captures error codes that are returned by the recognition device and converts these to easily understood text messages, such as "TOO LOUD" if the pilot spoke too loudly. Both the pilot and the experimenter receive this status information on their screens.

The Enrollment Program will present any sequence of words to the pilot, one at a time, for template testing. The program will also present short phrases of words for enrollment of these words in context. The selection of words and phrases can be made interactively by the experimenter or can be controlled from the Sequence File.

The Enrollment Program uses a technique developed by our colleagues Dave Williamson and Tim Barry at AFWAL/FIGR, Wright-Patterson Air Force Base, Ohio, and implemented in their Software Methodology for Automated Recognition Training (SMART)$_5$ Utility (Williamson, Small, and Feithans, 1987).^{5} Any particular word or phase for template enrollment is re-presented to the pilot in cases where the recognition device reports an unacceptable signal quality. However, an item is re-presented no more than two times. If the recognition device still fails to accept the input speech token, then this word or phrase is appended to the end of the enrollment sequence and repeated only after all other items have been presented to the pilot for enrollment. If a second series of attempts fails, then the enrollment program stops prompting the failed items. This procedure reduces pilot frustration and total enrollment time and ultimately provides better quality templates.

A text record of any or all segments of an enrollment or template testing session can be collected and saved on disk for later printout. Goodness of fit data from the recognition device on each recognition attempt are converted to numeric text strings and stored in the data file.

Templates for words can be deleted and new templates enrolled without altering the templates for other words. New enrollment sequences can be created interactively to enroll problem words in different phrase contexts.

Recognition Testing and Training

The Recognition program is used for testing the templates in context and for training the pilots on the command wording.

Template Testing in Command Wording Context. The Recognition program uses a previously edited text file, called the Syntax File, which assigns words to recognition sets and which specifies legal command wordings. The pilot is given a training manual with a list of the legal commands and uses this manual as a guide for testing the recognition of words in command string context. As a pilot speaks to the system, it recognizes word by word what has been said. Literal visual feedback of each word recognized is displayed on the pilot's color screen. The system also notifies the pilot via text display and, optionally, via spoken synthesized message, as soon as a word that makes an illegal sequence has been recognized. Meta commands, including DELETE and DISREGARD, can be spoken by the pilot to erase the misrecognized or mis-spoken words; then the pilot can say these words again. DELETE erases the last word in the recognized command string and may be used repeatedly until all words recognized for a given command have been erased. As a word is erased, it disappears from the literal visual feedback display. DISREGARD erases all words in the currently recognized command string.

When a command has been correctly spoken and recognized, the system emits a beep and is automatically initialized to receive another command.

Data collection can be enabled for any or all segments of a recognition testing session. The data is stored as a text file on disk for later printout. Included are the words recognized (in order of occurrence), the set number of that word, any occurrences of illegal command wording, and occurrences of meta commands.

Any words that are consistently misrecognized when spoken in context can be retested using the Enrollment Program and if necessary re-enrolled. The pilot can also elect to use a different pronunciation or a different word in place of the original word to better match his or her idiolect.

Pilot Command Wording Training. The same mode of the Recognition program can be used to train pilots in the correct command wording. A different training manual, which provides command wording practice, is given to the pilot. The pilot receives immediate feedback on errors from the "Say Again" synthesized speech message that is spoken any time an illegal command wording is recognized. And the beep issued upon completion of a correct command string provides positive reinforcement.

Pilot Command Wording Testing. Next, by means of a test booklet, this same mode can be used to test pilots' proficiency level of speech command use. The booklet contains memory tests for correct command wording. The pilot marks in the booklet any instances of misrecognition to distinguish these from cases in which he or she spoke the wrong word.

Cockpit Speech Command Procedures Training. The final mode of the Speech Recognition Program is a Speech Command Procedures Trainer. This mode is similar to the recognition training and testing modes described above. However, to it is added interactive spoken and/or text messages to the pilot as he completes each spoken command. For example, if the pilot says *Sight Control Left*,[**] the program responds: "Control of Mast-Mounted Sight is on the left". Messages can be spoken by the synthesizer or printed on the pilot's screen, according to an experimenter-edited text file called the Response File. Training dialogs with varying levels of complexity can be created to permit the pilot to interact via speech with this makebelieve cockpit.

SCR-Host Mode

It is customary in flight simulators for a single large computer to handle all aspects of the simulation. For those in which speech recognition is used, the actual recognition is done by dedicated hardware, but the simulation computer (simulator) still performs all interactions with the speech recognizer. The difficulty with this is that speech events are asynchronous with respect to whatever else the simulator may be doing. Although speech events occur at a relatively slow rate, nevertheless the response time of the simulator to them must be rapid (on the order of tenths of a second) to correctly identify sequences of command words and provide timely feedback to the pilot. If the simulator is performing some higher priority task when the speech event occurs, there may a considerable delay before it can take care of the speech input.

A simulator program has to be very large in order to handle flight aerodynamics, terrain mapping and control input and output. It is the work of many programmers and consequently is difficult to modify and subject to a large number of restrictions in so doing. In a simulator used for human factors research it is necessary that the computer-pilot interface be easily changeable. Just as a separate graphics processor offloads the handling of the visual display, the Smart Command Recognizer offloads the speech processing burden from the simulator, allowing faster response, fewer errors, easier modification of vocabulary and functions, and reducing processing time and program size on the simulator.

In the SCR-Host Mode, the SCR (with speech recognizer device) is interfaced to the host computer as a peripheral. This interface is through a standard RS-232C serial data bus. Handshaking between the host software and the SCR ensures that the communication will be successful regardless of which software is started first. All communications between the two consist of strings of printable ASCII characters.

The current application is a speech input system for a simulated Army helicopter in the U.S. Army's Crew Station Research and Development Facility at NASA-Ames. In this application, the helicopter simulation software resides in a VAX 8650 computer, which is interfaced to a fixed-base simulator cockpit. A GE CompuScene visual image generation system connected to a CAE Electronics helmet mounted display unit provides the visual out the window view to the pilot. In addition, three piloted "team station" consoles, incorporating MicroVAX computers and IRIS displays, can simulate friendly or hostile aircraft in the environment. A communication console with pre-recorded and live transmissions as well as speech disguisers enhances realism of the environment. In this system, the speech command input may be used by the pilot to activate complex switching sequences by command, or to input alphanumeric data to the computer, while keeping his hands on the aircraft controls.

SCR Side

The Smart Command Recognizer (SCR) is a microcomputer running a special program that controls the speech recognizer hardware via a serial communications port. The speech recognizer sends the codes for recognized words spoken by the pilot to the SCR. The SCR syntax interpreter software determines whether the word is a legal one in the context of previous words. Legal words are appended to the "command sequence" managed by the syntax interpreter. The command sequence is displayed visually to the pilot on the visual feedback display. Indication of illegal words is also shown on the visual feedback display. Thus the pilot is given constant immediate feedback on the status of his speech command. Most commands consist of a series of command words specified by a grammar for the particular set of words and parsed by the syntax interpreter. Both the words and the grammar can be easily changed by the researcher. There are commands (meta commands) by which the pilot can delete words from the command sequence or even abort the command entirely.

A second serial port connects the SCR to the simulator. All messages sent between the simulator and the SCR are in standard 7 bit ASCII characters. The first character of a transmitted string, chosen from the special characters in the ASCII character set (e.g., #, @, !), specifies how the rest of the string shall be interpreted by the receiver. For example, when the SCR wants the simulator to initiate a spoken message to the pilot with the speech synthesizer (such as "Say Again" when a syntax error is detected), it sends a 5 digit message number preceded by an identifier (!) and terminated with a carriage return character. The simulator may or may not actually deliver the message depending on the experimental conditions set by the researchers.

Verbal command sequences can give operational commands to the flight simulator, ask that certain information be displayed, or start the entry of data into the flight management hardware. When the completion of a command is detected by the syntax interpreter, the command number contributions of all the words in the command are added up and the sum sent to the simulator. The simulator then acts upon the command as it is programmed to do.

Before the command number is received by the simulator, it does not "know" that a verbal command is being processed. All functions

----

[**] The notation *Word* indicates that this word is actually spoken to the recognition device.

pertaining to that command are performed by the SCR. For example, if the pilot says the word corresponding to the meta command DELETE, the SCR will delete the last word from the command sequence, and from the visual feedback display, without informing the simulator that this has happened.

On the other hand, the simulator does pass on meta commands (manually entered by the pilot) to the SCR and the SCR treats them just as if the pilot had delivered them verbally. For example, if the pilot presses the key which activates the DELETE function, the simulator will send that meta command number to the SCR and the SCR will proceed to delete the last command word in the command sequence. The simulator, of course, does nothing else with the DELETE meta command since it has nothing to delete.

Some commands require alphanumeric data--a radio frequency setting, for example. The data can be entered both verbally through the SCR or via a keypad driven by the simulator. Both methods of entry can be used interchangeably and can even be intermixed. If the pilot says a data character, the SCR sends it immediately to the simulator, which adds the item to its data string and to the pilot's scratchpad display just as if he had entered it with the keypad. At present, the simulator does not send any data that the pilot enters using the simulator's keypad to the SCR, so the SCR only displays spoken data on its visual feedback display.

If the pilot speaks one of the meta commands, the SCR will perform the function and also send the command number to the simulator, since now both are involved in the inputting of data. Again using DELETE as an example, the SCR would delete the last data word spoken, and then send the DELETE command to the simulator which would erase the last entry from the cockpit scratchpad and from its internal data string. In this case the converse is also true: the simulator can send the DELETE meta command number to the SCR as well. The SCR will erase the last recognized word from the command sequence and the visual feedback display.

One of the meta commands is ENTER, which terminates a data string. It can either be spoken to the SCR or activated by a key press on the simulator. Whichever computer gets this signal informs the other so that data entry will be completed properly in both systems at the same time. Other meta commands operate in a similar way during data entry: both computers have to know that the command was issued.

Most of the commands that can be entered verbally can also be entered by function keys in the simulator. If a command that requires data entry is activated manually, the simulator sends a special command to the SCR requesting it to go into data entry mode. Then the simulator collects the data in parallel as described above. If it should happen that this data entry command comes in when the SCR is involved with a speech command, (remember that the simulator does not know if the SCR is processing a speech command) then it will automatically save the unfinished command and return to it when the data entry is finished.

There are two other meta commands (besides DELETE and ENTER which have already been mentioned) which the pilot may initiate. They are DISREGARD and CANCEL. These may originate either at the SCR or the simulator. The basic difference between them is that DISREGARD aborts an entire command while CANCEL merely erases all data input. On CANCEL, the simulator erases all scratchpad entries and the SCR removes all data entries from the visual feedback display. The pilot may then enter new data. On DISREGARD, the simulator will dump its internal command queue and the SCR will clear the the current verbal command and the readout display. Note that CANCEL is only applicable during data entry and that there is no need for the SCR to send the DISREGARD signal to the simulator except during data entry. The simulator, however, sends DISREGARD to the SCR anytime the pilot presses it, causing the SCR to abort any verbal command in progress.

The SCR keeps its own data file of recognized words, commands, meta commands and data for later analysis. Since all the data entered into this file are timestamped with the SCR computer's clock, synchronization with the simulation clock is necessary to compare related events. This is achieved by interlocking signals between the simulator and the SCR at the beginning of an experimental session. The simulator sends a MISSION START signal to the SCR when the simulation begins. The SCR responds with a TIME REQUEST command and then the simulator sends EXPERIMENT INFORMATION. The EXPERIMENT INFORMATION includes the current mission time, which is stored in the SCR's data file along with the SCR's equivalent computer time, thus providing the required synchronization. EXPERIMENT INFORMATION also includes a coded file name so that the SCR can determine that its input files are consistent with those being used by the simulator.

If the SCR program is not running when the simulator sends MISSION START, it will not, of course, receive the signal. This difficulty is circumvented by having the SCR automatically send the TIME REQUEST command when the program starts. If the mission is running at this time, the simulator will respond with EXPERIMENT INFORMATION, otherwise with NOT RUNNING. Of course, if the simulation program itself is not loaded, there will be no response at all. In the last case, the SCR will either wait for MISSION START or try again later, at the option of the experimenter. If the mission is NOT RUNNING, there is an additional option for the SCR to go online with the simulator to conduct tests of the equipment, program functions or command sequences.

This procedure insures synchronization even if one computer is not running when the other starts. Also, during the experiment, the SCR can be partially or fully disabled to simulate equipment failure.

Host Side

On the host side, communications received from the SCR fall into three categories: (1) Message numbers, (2) Command or Meta-Command numbers, or (3) Alphanumeric data. These categories are designated by the first character of the record sent by the SCR. This character is one of the following ASCII special characters: "!", "#", or "@", corresponding to the three catagories of information. If the communication is a message number, it corresponds to a verbal message to be spoken by the speech synthesizer, and is handled by being sent to the speech output software. If it is alphanumeric data, it is handled by being sent to the data input subsystem software, and is treated the same as data received from the keypad located in the cockpit. If it is a command number, then it is checked to determine if it corresponds to a meta command. If so it is handled immediately; otherwise it corresponds to a sequence of up to five primitive functions to be executed in sequence. This is accomplished as follows: a binary search is used to locate the command number within a table residing in the simulator's common database; the primitive functions associated with this command number in the table are loaded into a FIFO queue; then the functions are performed in sequence, one being executed at each call to the program by the simulation executive.

This system allows the flexibility required in a research environment. The researcher can change the primitive functions to be executed in response to a given command number at any time (even during simulator operation) without modifying the software. Changes to the table of correspondences are made by executing a separate detached process which reads a file containing the correspondences (in a user-friendly format, derived from the Set File), sorts them into ascending order of command numbers, and loads the table. The ascending order is required for the binary search, which is required because the list of command numbers is a sparse list produced, in the SCR, by a computational algorithm acting on the word numbers in the command phrase.

In addition to communication from the SCR to the host speech input software, some specific communications are sent by the host to the SCR. When the simulated mission is first started, the host software sends a code to the SCR to tell it that the mission has begun. When this is received, or when the SCR comes on line (whichever is first), the SCR sends a code to the host requesting the name of the file used to load the command/function table and the current time (which is needed in order to timestamp data collected by the SCR. Other host-to-SCR messages include such things as DISREGARD, DELETE or ENTER activated by pushbutton, requests to change the input gain of the speech recognizer, et cetera.

Speech Command System Development with the SCR

The SCR is being used in the Crew Station Research and Development Facility, U.S. Army Aviation Research and Technology Activity (Lypaczewski, Jones, and Voorhees, 1986),[2] and the NASA/Army TH-1S Cobra Voice I/O Experimental Flight System (Huff, Edgerton, and Wong, 1987)[1] at NASA Ames Research Center.

Design requirements for the SCR were first developed in November, 1985 and refined through April, 1986. The SCR functions were selected to be consistent with the planned capabilities for the Versatile Simulation Testbed for Rotorcraft Speech I/O System Design (Simpson, 1986),[3] a component of the U.S. Army's Crew Station Research and Development Facility (CSRDF). The CSRDF is a full mission combat helicopter simulator designed to support Army human factors research on advanced combat helicopter controls and displays. The SCR was used beginning in July, 1986 to develop vocabulary and command wording for spoken controls in the CSRDF. This development effort on the SCR was performed by Psycho-Linguistic Research Associates (PLRA) in parallel with, but 3500 miles away from the the site of, the actual building and programming of the simulator by CAE Electronics, the CSRDF Systems Integrator. In November 1986 the SCR was used to enroll templates and train Army helicopter pilots to use the CSRDF speech commands. These pilots then used the SCR-developed templates and speech commands during the first pilot evaluation of the CSRDF simulator at CAE in Montreal, before it was delivered to the Army.

Benefits of the SCR to Date

The SCR made it possible to develop speech commands, enroll templates, and train pilots to use the commands before the target CSRDF flight simulator had been built. This saved about six months of calendar time and about 40 hours of very expensive flight simulator time that would otherwise have been needed for template enrollment and speech command training for the eight evaluation pilots.

The SCR also was used to simulate the characteristics of an earlier version of the CSRDF Speech Command Syntax Interpreter as designed and implemented by CAE and to test its ease of use by the Evaluation Pilots. As reported earlier (Simpson & Krones, 1988),[4] development work with the SCR in stand-alone mode demonstrated that pilots, as well as researchers, tended to confuse meta words which were similar in meaning but not identical in function. As a result, the CSRDF speech command software is being modified to eliminate the confusing meta words. The versatility of the SCR Command Syntax Interpreter made it possible to easily modify the command syntax and test it before the simulator software was completed, again saving time and funds that would otherwise have been spent to discover the problem with these meta words later in the simulator during a full mission simulation experiment.

The SCR is currently being used to develop and test speech commands for control of cockpit systems including Communications, Navigation, Weapons, Battle Resource Management, and Aircraft Survivability Equipment. This application is critical to the high-workload environment of advanced Army helicopters, due to the need to keep the pilot's hands free for controlling the aircraft while allowing him to interact with the many cockpit systems required in today's tactical scenario.

Summary

The SCR is a combined development tool and an experimental test system. It has already saved many months of calendar time on a fast track Army simulator development project as well as a large sum of money in expensive simulator time. It is a working system that will be used repeatedly to support speech command design research in the Crew Station Research and Development Facility. The SCR concept has proven to be invaluable for the research environment, as it allows the flexibility required for speech input experimentation. In addition, it provides a standard interface for future speech recognizer hardware. bus.

References

1. Huff, Edward M., Edgerton, Millard, and Wong, Bruce, 1987. Development of a voice I/O experimental flight system for use in a TH-1S "Cobra" helicopter, Proceedings of AVIOS '87 Voice Input/Output Systems Applications Conference, Alexandria, VA, October 6-8, 1987. American Voice Input/Output Society: Palo Alto, CA.

2. Lypaczewski, Paul A., Jones, A. David, and Voorhees, James W., 1986. Simulation of an advanced scout attack helicopter for crew station studies, Proceedings of the 8th Interservice/ Industry Training System Conference, Salt Lake City, UT, 18-20 November, 1986. Sponsored by the National Security Industrial Association.

3. Simpson, Carol A., 1986. Versatile simulation testbed for rotorcraft speech I/O system design, 1986 Transactions of the SAE (SAE Technical Paper Series 861661), SAE: Warrendale, PA.

4. Simpson, Carol A. and Krones, Robert R. (1988). Smart Command Recognizer (SCR): A rapid prototyping system for speech commands, Proceedings of Speech Tech 88 Conference, pp. 67-70. Media Dimensions: New York.

5. Williamson, David T., Small, Ronald L., and Feithans, Gregory L., 1987. Speech technology in the Flight Dynamics Laboratory, Proceedings of the IEEE 1987 National Aerospace and Electronics Conference (NAECON) (87CH2450-5) Vol. 3, pp. 897-900. IEEE: New York, 1987.

Acknowledgements

The development of the SCR system design specification and the application of the SCR to the CSRDF speech command development effort reported here were performed by Psycho-Linguistic Research Associates (PLRA) for the Crew Station Research and Development Branch (CSRDB), Aircraft Simulation Systems Division, U.S. Army Aeroflightdynamics Laboratory, Moffett Field, California under NASA Ames Research Center Contract NAS2-12425. SCR software development was performed by Informatics Division, Sterling Software, also for the CSRDB, under NASA Support Services Contract NAS2-11555. VAX-8650 host software development was performed by SYRE, Inc. for the CSRDB, under NASA Support Services Contract NAS2-11631. We thank Major James W. Voorhees, PhD, Chief of CSRDB and Dr. Nancy Bucher, Director of Research, CSRDB and Contract Technical Monitor for PLRA for stimulating discussions and for providing the research and development environment that made this project possible.

THE USE OF AN EYE-TRACKING DEVICE FOR THE MEASUREMENT OF FLIGHT PERFORMANCE IN SIMULATORS

1st Lt Kevin W. Dixon
2d Lt Gretchen M. Krueger
2d Lt Victoria A. Rojas

88-4613-CP

Abstract

The most popular methods used to assess the design and training capabilities of a flight simulator are subjective questionnaires and objective performance measures. These types of methodologies suffer from their inability to provide accurate data concerning attention allocation and behavioral strategy. This added information closes the loop in data collection and would help validate the results of a number of studies that have focused on visual system characteristics. Visual attention information can be collected with an eye-tracking system. One eye-tracking system is described in this paper, along with its advantages and limitations, with respect to determining field-of-view requirements for visual systems. Other areas discussed include system architecture, initial implementation, and future applications.

Introduction

Designs and evaluations of simulator visual systems have historically depended on subjective questionnaires and objective pilot performance data to determine training effectiveness. The questionnaire approach uses groups of personal opinion data to evaluate characteristics of the visual system. One characteristic of the visual system is the field-of-view (FOV) requirements. The questionnaire approach has been widely used and accepted to answer the question of FOV requirements. However, questionnaires suffer from being subjective in nature, they give no indication of the portion of the FOV being used, and they do not reveal where attention is allocated. For example, an evaluation conducted by Goodyear-Redifusion for the F-15 Visual System asked 48 pilots to perform certain tasks in a 160° H X 60° V limited field-of-view F-15 simulator, rate the FOV for each task, and estimate additional FOV requirements for tasks rated less than acceptable. The results of this study, based on pilot opinion, state that the evaluated visual system (160° H X 60° V) can substantially enhance air superiority operations training and air-to-surface operations training. Subjective evaluations performed in this manner contribute a large amount of data; however, it is difficult to make quantitative and reliable decisions based on opinion and memory.

The use of pilot performance measures (altitude, airspeed, etc.) during the simulator mission can also be deceiving, because the pilot may use other means (instruments) to compensate for the lack of fidelity of the visual system. For example, a study by Irish and Buckland (1978) evaluating FOV requirements for the T-37 Advanced Simulator for Pilot Training (ASPT) found that pilots performed some tasks better with a wide field-of-view (300° H X 150° V), while other tasks such as a Ground Controlled Approach (GCA) were performed significantly better with a more limited field-of-view (144° H X 36° V). These contradictory results may be attributed to the fact that GCA's are instrument approaches, and therefore a wide field-of-view may be more distracting, since there is more information to interpret than in a limited field-of-view situation. However, these results do not necessarily mean that a limited field-of-view is better for GCA training, because the pilots' visual strategy in a limited field-of-view simulator may not be the same as the strategy used in the aircraft itself. Because pilots are taught not to use their visual cues during instrument approaches, it might be advantageous to teach an instrument crosscheck with a full field-of-view. Hence, the pilot will be familiar with the visual cues associated with the aircraft. However, visual behavior data cannot be captured using objective performance measures. This causes a gray area in the validity and reliability of such data.

Subjective questionnaires and objective pilot performance data give the researcher an idea of pilot impression and performance, but do not allow objective determinations of the pilots' visual activity. By continuously recording the pilots' visual activity, the eye-tracking system offers a new approach to quantify pilot visual perception during simulated flight. The eye-tracking system allows direct assessment of the visual behavior of the pilots, and where their attention is focused at any time during the mission. An eye-tracking system would help us understand how the pilots in the Goodyear-Redifusion study rated the field-of-view for each task as acceptable or unacceptable. It would also explain why the pilots in Irish and Buckland's FOV study performed GCAs better with a limited field-of-view. Perhaps this would validate the assumption that the subjects spent less time looking at the instruments, and more time looking at outside visual cues in the wide field-of-view situation.

The eye-tracking system also has its limitations. It only reveals focal point and does not deal with visual information that is available and processed from the visual periphery. For example, when the pilot is looking in a forward window, his periphery is also being stimulated by the information in the adjacent window. Although he does not look at the adjacent window as frequently as the forward windows, he is often getting input from this window, especially when the focal point is in the forward window. Despite the potential limitations, the advantages of having focal point data outweigh the limitations.

System Architecture

After determining that eye position data would help fill the gap between subjective and objective data, a suitable system to collect data on visual behavior had to be found. A number of factors contributed to the selection of an eye- tracking device. The major factors were cost, ease of calibration, comfort, and free head movement. The system chosen was a Model 210 eye movement monitor from Applied Science Laboratories. The Eye Movement Monitor employs a photoelectric sensing and processing technique to determine magnitude and direction of eye movements. Eye illumination and sensing are accomplished with infared illumination to minimize distraction to the subject. The device is attached to a head-band mounted camera (See Figure 1) which provides a video fixation point capability.

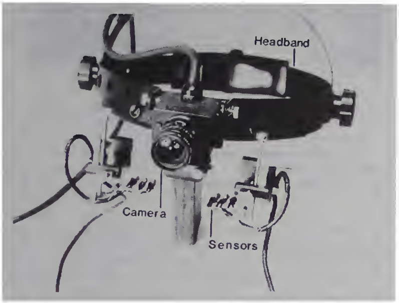


Figure 1: Eye-Monitoring Apparatus

The instrument is capable of measuring horizontal eye movements over a range of approximately +/- 15 degrees with an accuracy of about 1 degree and a precision of better than 1/4 degree. Vertical eye movements can be measured over a range of approximately +/- 15 degrees with an accuracy of about 2 degrees and a precision of better than 1 degree (See App A, System Specifications). By allowing free head movement, the device can contain an instantaneous field-of-view of 30 degrees horizontally and vertically, and a full 360 degree field-of-regard. The only limitation is for tracking eye movements that are out of the instantaneous field-of-view.

After finding the appropriate eye-tracking device, the data recording procedure was investigated. The video fixation point capabilities of the device present either cross hairs or a cursor superimposed over a television monitor image of the scene being viewed by the pilot. The image is captured with a video recorder and time coded to complete the data collection procedure. The last component of the system is a computer software program to analyze the data. Tapemaster was used to code the collected data. The Tapemaster program allows area definitions and associated time code stamps to be collected. This data is further analyzed to give descriptive and inferential statistics of the data. Figure 2 is a diagram of the overall system.

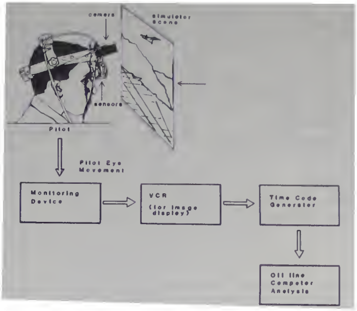


Figure 2: Diagram of Eye-Tracking System

Implementation

Preliminary evaluations were conducted to insure component compatibility and proper system operation. Familiarization and proficiency training was conducted in a static environment to practice set up and calibration technique.

The initial field test of the system was accomplished during a recent C-130 Weapons System Trainer (WST) field-of-view study at Little Rock AFB, AK. The C-130 WST is a full mission simulator which provides computer-generated imagery for out-of-the-window visual cues. The visual system produces day, dusk, and night scenes through a six window, five channel, color CRT display system with infinity optics. This study investigated the effect of field-of-view (FOV) on pilot performance for low level flight and an airdrop in the C-130 weapon system trainer. The study was performed using two different field-of-view configurations. The wide field-of-view condition incorporated all six windows to provide a visual field of 160° H by 35° V. The limited field-of-view consisted of the forward four windows to provide a 113° H by 35° V visual field.

Two methods of data collection were used throughout the study. The first method incorporated the eye-tracking system to determine whether or not pilots' visual behavior or performance is altered in the different FOV configurations. Automated pilot performance measures were also collected and included pilot control inputs and system parameters. Twelve male C-130 pilots with a crew qualification of instructor pilot or aircraft commander served as subjects. The eye-monitor device was worn by the pilots for the duration of the simulator test flight which lasted approximately 25 minutes. Subjective questionnaires were given to each pilot at the conclusion of the test. The pilots reported only slight discomfort after wearing the apparatus, but reported that it did not restrain head movement or interfere with the mission.

Data from the eye-position camera were encoded using a personal computer applications program. This program (Tapemaster, Comprehensive Video Supply Corp.) enabled the specification of visual area codes (area within each window, instruments or other) for the visual field based on the video fixation point. (See Figures 3, 4, and 5) "Instruments" were defined as eyes transitioning to the instrument panel and "other" was all actions not related to windows or instruments. The definitions for each area were encoded into the computer by hand. Once encoded, the data were transferred to the VAX 11/780 for further analysis. The variables used for analysis included: time in each window, glances in each window, percent of total time and glances for each window and percent of time per glances.

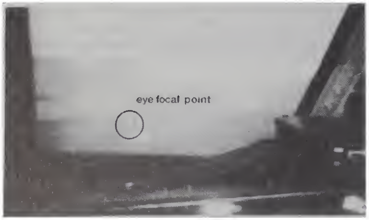


Figure 3: Example of C-130 Eye-Focal Point (Front Window)

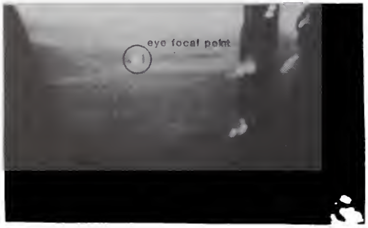


Figure 4: Example of C-130 Eye-Focal Point (Side Window)

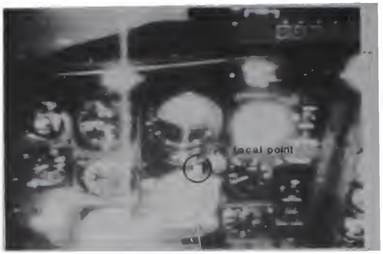


Figure 5: Example of C-130 Eye-Focal Point (Instruments)

The pilot performance data showed no strong or consistent effects due to the field-of-view manipulations. However, the results from the eye position data revealed an increased use of the front window and instruments in the limited field-of-view condition and a decreased use of the window directly to the left of the pilot. The study shows that the peripheral windows may not be required for experienced pilots, but if present, they are used, and if absent, alter visual behavior. While the pilots tended to use their peripheral vision during the full field-of-view configuration, they used their instruments more during the limited field-of-view configuration. This is probably due to the loss of outside visual information.

This initial test confirmed overall system performance and operation in a dynamic environment. The use of the eye position data in field-of-view research is clearly an advancement over relying on ratings and objective performance data from the system state of the aircraft. Having direct access to the pilots' visual behavior allows for a much greater in depth understanding of the impact of the variable.

A second field-of-view study was done on the Simulator for Air-to-Air Combat (SAAC) located at Luke AFB, AZ. The objective of this study was to determine the field-of-view used in specific Air-to-Air tasks, free engagements, and mutual support operations. The simulator facility consists of two F-15 cockpits and a computer interface that allows the pilots to train against each other or against the computer in air combat engagements. Data on number of glances in each window, time spent on instruments, number of transitions from each window, and eye position relative to target position were collected. Final data analysis is currently underway and the results will be published at a later date. In contrast to the relative stability in the C-130 simulator, air-to-air combat maneuvers require a great deal of extraneous head movement. The incorporation of the eye-tracking system in this environment was a significant step in validating system versatility and flexibility.

Applications

Results from the static and dynamic tests of the eye-monitoring system indicate strong potential in both research and training environments. Proposed visual training research applications include areas in workload assessment and attention allocation. Many of these applications require initial research into visual behavior and its relationship to mental processing. For example, workload assessment cannot be done until adequate investigations are made to determine the exact relationship between visual behavior and mental workload. If tests indicate a definite correlation between workload and visual attention, the eye-tracking system will prove invaluable for workload assessment.

Another area which holds potential for the use of the eye-tracking system is training research. Investigations into such areas as pilot crosscheck techniques is just one of the many training applications possible. By comparing the visual strategy of individual pilots (focal point, time spent on each instrument, etc.) recommendations for crosscheck training for inexperienced pilots could be made.

The eye-tracking system could also be used for system engineering design and evaluation. For instance, a quantitative evaluation of the visual cues used in low level flight can be obtained with the eye-tracking system. Such an evaluation will be extremely useful in order to optimize scene content. If incorporated into research and design methodologies, the eye-tracking system can provide valuable information on the eye focal point for various tasks. The advantages of the eye-tracking system make it a valuable tool that can be used alone or in conjunction with subjective and objective performance measures for valid and reliable decisions concerning visual systems.

References

Applied Science Laboratories. Model 210 Eye Movement Monitor: Operation and Service Manual. May 1984.

Comprehensive Video Supply Corp. Tapemaster 1.0: Owners Manual and Reference Guide. 1986.

Dixon, K.W., E.L. Martin, Phd., V.A. Rojas, D.C. Hubbard, Phd. (July 1988). The Effects of Field-of-View on Pilot Performance in the C-130 WST (AFHRL - unpublished). Williams AFB, AZ: Operational Training Division, Air Force Human Resources Laboratory.

Irish, P.A., III, and Buckland, G.H. (June 1978). Effects of platform motion, visual and G-seat factors upon experienced pilot performance in the simulator (AFHRL-TR-78-9, AD A-055 691). Williams AFB, AZ: Flying Training Division, Air Force Human Resources Laboratory.

O'Neal, Maston E. (July 1984). F-15 Limited field-of-view visual system training effectiveness evaluation. (TAC 83G-066T). Langley AFB, VA: HQ TAC.

APPENDIX A
SYSTEM SPECIFICATIONS

	HORIZONTAL	VERTICAL	REMARKS
Eye Movement Range (from Center)	±15°	±15°	
Precisions (see Section of Section 1.7)	0.25°	1°	A few minutes of arc possible with rigid head restraint.
Accuracy	1°	2°	
Response Time Constant	unfiltered - 4 milliseconds filtered - 26 milliseconds		
Outputs Signals (Both Channels Simultaneously Available) (see Section 4.0.)			
Analog	300mV/degree (nominal) ± 3V maximum		
Digital	3 bit binary for each channel DTL/TTL Compatible Offset binary code		Both digital outputs updated once each millisecond; busy-bit signal during updating (see Section 4.0)
Instrument Drift	10mV/hour		
Calibration Controls	Electronic position, gain, linearity and crosstalk controls are provided.		Crosstalk is usually adjustable to less than 10%
Power Requirements	105-125V AC, 50-60 Hz 25 WATTS		230-250V AC
Weight	8 lbs.		
Dimensions of Control Unit	12.5"W x 11"L x 5.5" H (31.25 cm x 28 cm x 14 cm)		
Artifacts	Blinks (readily distinguished), Squinting		

SIMULATION TOOLS FOR CREW SYSTEM ASSESSMENT

88-4614-CP

*Brett A. Storey**

Lockheed Aeronautical Systems Company

P. O. Box 551, Burbank, CA

ABSTRACT

This paper is intended to highlight the need for a greater concentration of effort into data gathering methods and data analysis tools. Too often in today's high technology, high fidelity simulation environment, all effort is directed toward making a simulation as real as possible while perhaps forgetting the very reason we do testing. That is to gather and analyze data both qualitatively and quantitatively. The crew system design analyst needs to understand more about what went on than "I liked it," not that I liked is bad, it's just that more concrete data is needed to understand what each pilot can and cannot use to his advantage.

Specifically, this paper will address the type of data needed, how to meaningfully display it to designers and how to provide an intelligent way to analyze it before reducing the data to statistics.

INTRODUCTION

This report describes a methodology of simulation research tools which are designed to accomplish the objectives for assessing the effectiveness of a crew system for fighter aircraft.

The specific objective in a crew system assessment program is to perform a comprehensive evaluation of cockpit control - display integration, pilot procedures/tactics, mission scenarios, pilot workload, situation awareness and system effectiveness. In order to achieve this objective, several components must be on board in addition to state-of-the-art simulation equipment. These components, while not always considered to be primary, are really the key to assessing cockpit effectiveness. This "better half" of simulation is data. What data to take, when and how to display it, and how to analyze it are the questions most crew system designers and analysts struggle with while making design decisions. These decisions need to be made with objective man-in-the-loop simulation data supporting the pilot subjective and designer analytical data.

NECESSARY DATA

The type of data needed can be divided into three separate categories: (1) subjective measures, (2) primary task measures, and (3) physiological measures. Each of these measures taken alone can provide some insight into what is going on with the pilot, but the combination of these measures can provide a means for measuring your design against a previous one.

Subjective

One method for subjectively evaluating pilot workload is the Subjective Workload Assessment Technique (SWAT). SWAT is a procedure by which pilot opinion of workload is obtained. SWAT deals with three dimensions of workload: time load, mental effort, and psychological stress. (A complete definition can be obtained in SWAT Documentation.) A pilot's relative weighting of these factors will affect his overall workload rating.

SWAT ratings are collected for general tasks or grouping of tasks (i.e., the action of launching a missile may require four distinct pilot actions), but not to the level of individual switch actions. Pilots are provided the definitions for each SWAT factor and are asked to provide a single number for each factor. These responses can be captured verbally following the completion of each requested task sequence.

Other more common methods to obtain subjective data are through the use of either carefully structured rating forms and/or questionnaires or by allowing the pilot to replay his mission events and comment "free flow" about what his perception of the events were.

Primary Tasks

Primary task measures can be broken down into three separate categories including (1) pilot performance data, (2) aircraft performance data, and (3) mission effectiveness data.

Pilot Performance Data. Pilot performance can be recorded in the form of control actuations. The control actuations can provide information on the physical performance of each pilot. Each control input by the pilot, (whether it be by Hands on Throttle and Stick (HOTAS), dedicated switches, finger-on-glass, or voice) should be recorded by location versus time.

This record can be used to build a plot showing pilot task complexity versus time for significant sections of test scenarios. These plots should show mission time-correlated task printouts on which every pilot action is identified with the particular equipment and/or method of operation involved.

Aircraft Performance Data. In order to truly analyze aircraft performance data, it must be viewed in both a digital and graphical form. Data from each of the following categories must be used in order to evaluate the crew system design.

Takeoff and Departure
Navigation
Air-to-Air Combat
Approach
Landing

A. **Takeoff and Departure.** The following data should be recorded during the Takeoff and Departure phases:

- Takeoff time
- Takeoff distance (start to lift-off)
- Rotation Speed (nose wheel lift off)
- Average Rotation Angle (pitch angle from rotation to lift off)
- Takeoff Speed (speed at lift off)

[*] Lead Engineer, Crew System Assessment Member AIAA

- Gear up Conditions (gear initiated)

  - Pitch, bank, speed, altitude, rate of climb

- Flight conditions and error from departure plan at up to six waypoints (store at time of closed approach in ground plane)

- Pitch, bank, speed, altitude, rate of climb, time, heading ground track

- Altitude error, ground track error, distance at closest approach, speed error

B. **Navigation.** The first departure point should be coincident with the last departure point for quick look purposes. The following leg and turn point information should be based on closest approach to a turnpoint.

Turn Point Data	**Leg Data**
Altitude (MSL)	Average Altitude Error
Speed	Average Speed Error
Magnetic Heading	Average Heading Error
Magnetic Ground Track	Average Ground Track Deviation
Time	

Average information should be tracked in three ways: average absolute value of error, average error above/left of desired value, and average error below/right of desired value.

C. **Air-to-Air Combat.** For each engagement in an air-to-air scenario, timeline data can be provided showing target spatial relationship to the aircraft in the scenario. The order of events may vary from that presented and all events may not necessarily be present (e.g., missile launches are not attempted on all targets).

Air-to-air missile and air-to-air guns firing conditions can also be captured for use in the evaluations.

Air-to-Air Missile Firing Conditions include the following:

Firing Time
Target (Automatic)
Weapon Selected
Firing Range - (Ft)
Initial Closure - (KTAS)
Initial Target Aspect
Predicted Missile Time of Flight - (Sec)
Target Bearing Angle Off Nose
MLE Status at Launch
R_{1max}, R_{2max}, R_{1min}, R_{2min} (nm)

Tgt "G" at Impact Time
Tgt Countermeasure & Times (after launch)
Chaff - 1 Sec ECM On - 15 Sec
Flare - 3 Sec
Flare - 10 Sec
Target Track
Master Arm Status (SAFE, ARM)
Missile Parameter
Miss Distance and P_k
Average Target G from Launch to Impact

Air-to-Air Guns Firing Conditions should include the following:

Initial Bullet TOF
Target (Automatic)
Mode
Firing Range - (Ft)
Burst Length - (Sec) and (No. of Rounds)
Settling Error - (Mils) and (Feet)
Pipper Average Error
Total - (Mils) and (Feet)
Azimuth - (Mils) and (Feet)
Elevation - (Mils) and (Feet)
Master Arm Status
Cease Fire Range - (Feet)
Initial Closure - (KTAS)
Initial Target Aspect
Target Track

D. **Approach.** Approach data recording for interpretation will begin at the Final Approach Fix and terminate at touch down. Data required for a two view pictorial representation of the approach is listed below:

- ILS/LOC channels and when selected
- MLS channels and when selected
- TACAN channels and when selected
- UHF and VHF channels and when selected
- Gear down time
- Glide Slope Errors (ft)

  - Average Absolute
  - Average Above
  - Average Below
  - Maximum Deviation

- Course Errors (ft)

  - Average Absolute
  - Average Above
  - Average Below
  - Maximum Deviation

E. **Landing.** Landing data should include the following:

- Landing Time
- Touch Down Speed - (KCAS, Gnd Speed)
- Touch Down AOA
- Touch Down Sink Rate (Ft/Sec)
- Touch Down Drift (Ft/Sec)
- Touch Down Heading
- Touch Down Position/Deviation From Touchdown Aimpoint

  - From Approach End - Ft
  - Right or Left of Center Line - Ft
  - Pitch and Bank

Additionally, the following pictorial formats are required to be available for review on a CRT (during a test) and a hard copy following a test.

A. **Aerial Combat Profile**

An example of an aerial combat profile (Figure 1) depicts a spatial trace of the flight path of an fighter during the trial run.

The trace should be presented with the fighter starting from the left side. The viewing eyepoint should be high and offset so that the depth of the depiction will be on a cardinal heading or orthogonal to the longest "straight" navigation leg. The trace(s) should originate at initial run conditions and

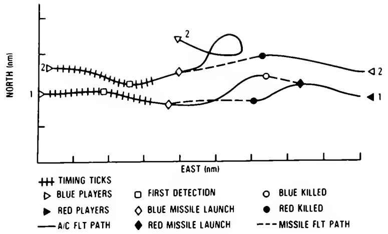


Figure 1.

terminate with aircraft kill or run termination. Major events should be symbolically depicted on the profile with a legend annotating their meaning on the format. In small scale scenarios (1 vs 1), all aircraft should have flight path/missile traces. Initial and final conditions as well as significant events in between should be symbolically depicted for all other participants. Route timing should be indicated by timing ticks with preset intervals.

B. God's Eye View Profile

The God's eye view format (Figure 2) should depict spatial relationships of participants in the tactical scenario from above.

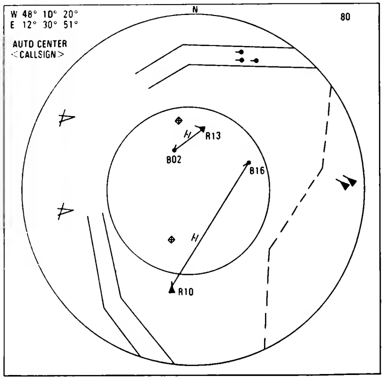


Figure 2.

The display should initialize in a North-up orientation but should have the ability to rotate a full 360 degrees to any orientation. The display diameter should also be selectable from 240, 160, 80, 40, 20, 10 and 5 nautical miles. A number of aircraft can be designated for a "centroid shift" function. When the function is commanded, the horizontal relationship between these aircraft should shift to the center of the display.

Surface symbology such as SAM sites, steer points, the FEBA, and airfields should be available for display. Aircraft and missile flight path traces for selected participants should also be available. Aircraft pitch and roll maneuvers should also be graphically displayed.

C. Navigation Profile

A navigation profile is similar to the aerial combat profile in that it depicts a historic trace of the aircraft as flown in a trail. This format can be used primarily for air-to-surface scenarios with either one or two aircraft.

The major differences between the navigation profile and the aerial combat profile are the desired flight path trace and symbolic depictions. The navigation profile includes a representation of the planned or desired flight path for view of deviations or excursion experienced during the trial run. The symbology and accompanying legend should have information of interest in the ground attack role such as Surface-to-Air Missile (SAM) engagement zones, steer points, target concentrations, and FEBA.

D. SAM Engagement Proflle

The SAM engagement/turn point is a magnified two-view of the maneuver implemented by the pilot and the SAM (if fired). The actual flight paths of the aircraft and the SAM in both horizontal and vertical flight path in delta excursions should be shown.

E. Approach Two-View

The approach two-view is a magnified view of the final portion of the navigation profile. The depiction, as shown in the example in Figure 3 shows desired and actual flight paths from one minute prior to the Final Approach Fix (FAF) to touch down.

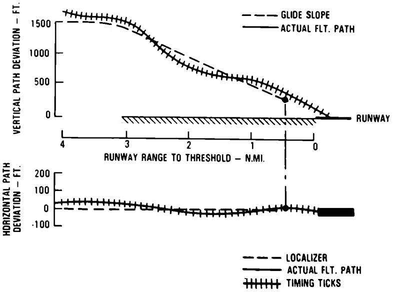


Figure 3.

<u>Mlsslon Effectiveness Data</u>. Measures of Effectiveness (MOE) data are derived from the results of the air battle. Measures such as number of aircraft destroyed or missiles launched vs. missiles registering kills, can be used to correlate how well the pilots performed individually vs. the success of the mission. This data is used with the pilot performance, subjective, and physiological data to gather a global view of how the cockpit design may help or hinder the pilot.

Physiological Measures. Physiological measures of workload can be obtained to provide an indication of the physiological cost incurred by a pilot in meeting task demands. The fundamental concept is that human physiological processes, such as nervous system activity, circulatory system activity, respiratory activity and body fluid chemistry, undergo involuntary changes in response to changes in workload. While this is not a fully proven method of detecting workload, it has been given increasingly more attention as the field data is correlated. The U.S.A.F. AAMRL scientists have developed a battery of tests to use in the measurement of human physiology.

DATA REVIEW

A full mission simulator should be able to acquire, monitor, process and playback sampled performance data in digital, graphic and audio form from all sources (simulator cockpits, threat stations). This data may be used for design analysis, debriefing and verification/validation purposes.

Data Consoles. Two sets of consoles (in addition to a simulation control console) need to be present in order to fully analyze data. These include an observer's console and a data monitor console.

The observer's console provides the primary capability to monitor the information presented to the pilot in the FMS cockpit environment. The console should be configured to allow the test team to simultaneously monitor all cockpit displays available to the pilot.

The data monitor console should provide display screens plus controls to select and adjust the data formats during simulation tests. The console can be used to display a variety of real-time data and information not available at the observer's console. This information enables the test team to monitor aircraft performance, system status, pilot response and the overall simulation environment. The console should provide the capability to select and adjust the display formats during the simulation. As a minimum, the formats should include digital readout of selected data, pictoral formats and delta X,Y,Z parameters, time plots, and a 3-D view of the air battle.

Data Features. During any formal data collection exercise, the following features should be available to review data for analysis.

Real Time Data Monitoring. A simulator must be capable of monitoring pre-specified data in real time. Data to be monitored should be specified at initialization. The monitoring of data must be flexible enough to allow for the collection of difference parameters at various capture times. These parameters should be displayed both in a tabular (digital readout) and graphical (2D and 3D) form. Additionally, a hardcopy or "snapshot" of selected parameters should be produced.

Quick Look Data Review. Quick look data in one form or another is required for all data runs. Quick look data provides the test director, test subjects, and test observers with short term feedback. Validity of the trial run and quality of data is verified using quick look data. Three basic categories of quick look data are required: pictoral formats (both 2-D and 3-D data), audio/video replay, and quantified/state data (pilot and aircraft performance, and MOE's).

Scenario Playback. The capability for digital recording of the simulation run for playback is required. The test director should be able to locate and playback selected portions of recorded data and view the displays in real time from the observer and data monitor consoles. Probably the most important attribute, playback of the simulation, may be used in the reconstruction of the run for analysis by both the test director and subject during debriefing.

ANALYSIS TOOLS

Simulation analysis tools must be capable of supporting quick turn around and edit features. The user interface needs to include a high resolution raster display. This display needs to be equipped with with a keyboard, touch screen and mouse for interactive user control. This analysis system should be tailored to produce usable results from large amounts of data which could result from a single fighter mission.

The emphasis on the analysis tools should be flexibility of edit capability. A analyst may have to sort through mounds of data in order to finally find what he is looking for. If the system software is flexible enough, this can be done shortly after each test run to ensure the continuity from the test run. The ideal software would allow for the graphical and digital data to be "scrolled" side by side on monitors to find the areas of interest for the objectives of the test. Data then can be marked from beginning to end of each event. In this way, data can then be sent to an analysis package with an assurance of what will result. As always, a standard statistical package such as the Statistical Analysis System (SAS) can be hosted on the same computer as the playback and edit features to comprise one single analysis tool.

SUMMARY

In order to fully exploit the data taken during full mission simulation, a set of data analysis tools need to be implemented within the simulation environment. The proper data must be taken (both from the pilot and the aircraft), that data must be displayed meaningfully to the analyst (both in digital and graphic forms), and the crew system designer/analyst must have the ability to look at all available data in a time correlated sequence. By allowing the capability to examine all of these data features, the particular events of interest can be focused on and thoroughly evaluated.

REFERENCES

1. Reid, Gary B., Eggemeir, F. Thomas, and Shingledecker, Clark A. Subjective Workload Assessment Technique, Proceedings of the Workshop on Flight Testing to Identify Pilot Workload and Pilot Dynamics 1982, Pgs. 281-287.

2. Shingledecker, Clark A., A Task Battery for Applied Human Performance Assessemnt Research, AFAMRL-TR-84-071 Air Force Aerospace Medical Research laboratory.

3. Storey, Brett A., Human Factors Engineering Simulation Methodology, Proceedings of the Human Factors Society 30th Annual Meeting 1986, Pgs. 1306-1310.

4. Storey, Brett A., Kurihara, Rod A., ATF Test Requirements Document for full mission simulation. Lockheed Aeronautical Systems Company, LD50-0127 1987.

5. Storey, Brett A., ATF Human Engineering Dynamic Simulation Plan, Lockheed Aeronautical Systems Company, LD50-0033, 1987.

6. Williams, Amanda M., Thomas, Gary S., A Research Tool to Improve the Effectiveness of performance measurement within the IOS Proceedings of the 9th Interservice/Industry Training System Conference 1987, Pgs. 117-120.

88-4616-CP

PROTOTYPE DATA ACQUISITION AND ANALYSIS SYSTEM FOR NAVY OPERATIONAL FLIGHT SIMULATORS *

R.C. Muller, G.O. Allgood, and B. Van Hoy
Oak Ridge National Laboratory†
Oak Ridge, Tennessee

Abstract

A problem of flight simulators has been the discomfort experienced by some pilots to the point of nausea. Likely explanations are a significant lack of synchronization between sight and movement as well as motion in the critical frequency magnitude region. A program to examine this problem has been undertaken at Oak Ridge National Laboratory, and the selection of appropriate computer hardware and software for analyzing motion and visual systems in real time is described. While requirements such as high data acquisition rate and high rates of arithmetic operations have driven the selections, rapid advances in computer technology have guided system development toward easy upgrade at modest cost. Results from use and demonstration have been positive, especially in the areas of reliability and ease of use.

I. Background and Purpose of System

Flight simulators are a very important aspect of pilot training. They provide a very cost-effective and increasingly realistic method of readying a pilot for operation of actual aircraft. They not only provide basic training for pilots, but training in tactics, emergency procedures, instrumentation, and virtually every facet of flying. Economy and safety thus achieved are excellent.

Because of these diverse uses of a simulator, pilots often have need to use one even after they have had considerable experience in the actual aircraft. Included in this type of operation is emergency procedure requalification, a situation in which the safety factor of the simulator is outstanding when one considers the problems encountered in duplicating such a procedure in a real aircraft.

Modern simulators attempt to duplicate both the motion and visual presentation of the aircraft. The tendency is toward increased fidelity in all modes of awareness. Obviously this is no small task, and modern simulators represent a large capital expense.

These very significant benefits do not come without some cost. A part of this is that some pilots tend to feel discomfort in the simulator that they do not experience in the aircraft. Analysis of simulator data has shown that this discomfort is often most keenly felt in pilots who come for postflight training after spending considerable time in real aircraft. In some cases the discomfort results in nausea and vomiting, typical of motion sickness. When we say motion sickness we are including visual effects, since the two are tied closely. Naturally this discomfort makes for loss of training effectiveness in addition to the negative aspects of the discomfort itself.

The problem then is to find the reasons for the discomfort. This involves psychological as well as physiological studies. MILSTD 1472C sets forth magnitude vs frequency curves for people who experience sickness for a certain duration of motion. Figure 1 shows the curves of this standard along with a number of typical situations in the ordinary world of motion experience. Many humans commonly experience motion of one sort or another, but normally it is outside the critical frequency-magnitude region shown in the figure. Of course, the studies of the human being must be done in conjunction with analysis of the simulator itself, and the system that we have assembled is for that purpose. Analysis of both motion and visual effects is needed.

Several aspects of the latter need to be investigated. Possibly the most important is the latency of the visual presentation with respect to the actual motion. Significant lack of synchronization between these gives the brain information that it does not like, possibly causing the discomfort that we have spoken about. Additionally, flicker and a jumpy picture could cause the same. It is known that brain waves normally beat in certain fixed frequencies and that visual frequencies near those can disturb the pattern in some individuals.

II. Choice of Hardware and Operating System

A good bit of time was spent in deciding upon the main supplier of the system needed to facilitate our requirements. A large number of manufacturers are available who supply such hardware, allowing for a great variety of systems from which to choose. However, because of our needs, this number was soon reduced considerably. Our requirement to look at visual systems in real time, which necessitates high data acquisition rates and high rates of arithmetic operations, meant that we would be pushing the limits of computer hardware from both speed and memory considerations.

The VME bus structure was chosen for several reasons. The accommodation of modern 32-bit architecture with its high throughput rate and the provision of a large variety of available off-the-shelf modular hardware are two of them. The latter results in the ability to do a phased

[*] Research performed under Data Systems Research and Development Program in support of the Naval Air Systems Command under Interagency Agreement No. 1682-1682-A1 and performed at Oak Ridge National Laboratory.

[†] Operated by Martin Marietta Energy Systems, Inc., for the U.S. Department of Energy under Contract No. DE-AC05-84OR21400.

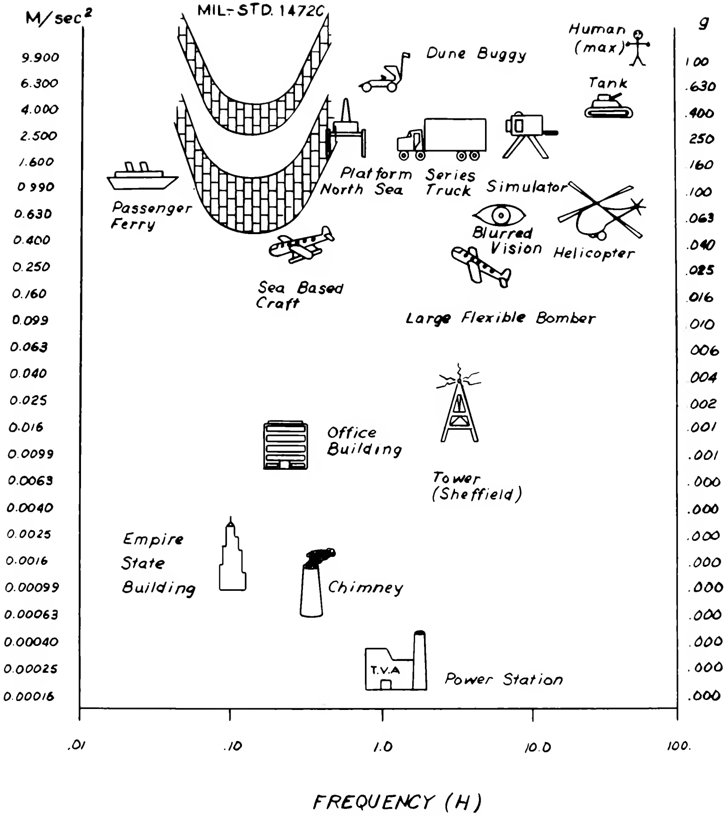


Fig. 1. RELATIVE VIBRATION MAGNITUDES

implementation rather than having to have a complete knowledge of all requirements at the inception of the project. Looking a year down the road is often quite difficult, and being able to meet demands as they come is a large benefit. Another valuable asset of the VME bus structure is its relative ruggedness compared with other buses, due in large measure to the connectors which are used. For this reason VME has gained an acceptance with the military that other buses have not.

One of the main tools of our motion dynamic analysis is the fast Fourier transform (FFT), which uses trigonometric functions. This means that a good floating point coprocessor is important for this and for many other functions where high computational throughputs are necessary.

We realized in the beginning that a real-time operating system was needed for most of the data taking. A reason for this is that the Fourier analysis mentioned above depends upon taking data at precise, fixed intervals. An operating system, such as UNIX, which does not have good real-time capabilities, would quite likely cause the system to miss clock pulses, thus causing the analysis to be suspect.

These are some of the reasons for choosing the system that we did and for selecting FORCE as the major supplier. FORCE has a good variety of modern hardware to choose from, as well as a close working relationship with TECHNICAL SYSTEMS CONSULTANTS, a supplier of real-time operating systems (RTOs). Their central processing unit (CPU) boards include those that use the 68020 processor and the 68881 coprocessor as well as

UNIFLEX, the RTOs written for them. At the time we were purchasing, FORCE was one of the few manufacturers that utilized these advancements. Now, with the breakneck pace of the computer industry, new CPU integrated circuits such as the 68030 and the even newer reduced instruction set computer (RISC) processors are becoming available. Naturally it takes a little while to integrate these into both the hardware and software portions of a system. When these improvements become available, however, it will be possible to upgrade our systems for a modest cost, at the same time reducing our board count with the advent of improved memory and glue chips.

The operating system we chose to purchase does not have memory management with it; but certainly for our system, the advantages of this situation outweigh the drawbacks. The capability to freely access any address is important for such operations as data exchange between concurrent tasks and the reading of such hardware as real-time clocks and data acquisition boards. Some speed increase also accrues for lack of memory management. However, software quality assurance needs are increased due to the ease of marring the system with undetected faulty addresses.

For the motion analysis we purchased DATEL analog-to-digital data conversion boards. Their conversion speed varies from about 5 to 100 microseconds, depending upon the gain that is set. This is adequate for our needs, even when sampling a dozen or more channels, since we normally do not need to take more than 50 samples per second.

The gain of the converter is programmable but not individually; one gain setting serves all channels. If we were to ask for an improvement, programmable gain adjustment for individual channels would be high on our list. At first glance it would seem that this could be done without needing A to D converters for each channel.

DATACUBE boards have been purchased for the acquisition of frames of video data, but we have not yet had opportunity to put them to use.

Figure 2 is a diagram of the system as we have used it to date.

III. Capabilities, Advantages, and Disadvantages of the System

The advantages of a computer system include, of course, the versatility to do special purpose as well general purpose analysis of data. By general purpose analysis we mean that which might be done on a standard Hewlett-Packard analyzer such as FFT, transfer function, and the like. With a system such as we have designed, we have the capability to do specialty items such as "activity number," the measure of energy at critical human frequencies that is absorbed by the human body and that can cause the discomfort with which we are dealing.

The multiprocessing capability that we have allows us to do several things simultaneously, especially to allow one processor to be occupied solely with data acquisition and another to do arithmetic operations and plotting. A common "phys" section of memory reserved to pass data and other parameters makes for rapid signaling and data transfer, more rapid than could be done by such built-in mechanisms as message passing and the like. This simple form of parallel processing has gone quite well without encountering problems of any significance. Figure 3 shows a diagram of our use of multiprocessing.

The attention we paid to hardware speed seems to be well repaid when we do such items as FFTs. We present a short table of our FFT results here.

Double precision fast Fourier transform results

Number of points in FFT	Time in seconds
2048	0.73
4096	1.56
8192	3.35

QUANTITATIVE TECHNOLOGY CORPORATION software was used in these measurements.

In order to utilize the ruggedness of the system in an operational situation, EEPROM must be substituted for the hard disk, the latter being somewhat sensitive to vibration. For this reason a board was ordered with our system for installation of these chips. This will contain enough of the operating system and needed programs to do the data acquisition. The data may be stored in battery backed SRAM, of which we have 2 megabytes.

In our previous experience with data gathering, we have used analog FM tape recorders. These leave a great deal to be desired in ease of use and in signal-to-noise ratio. Often, if the signal is lower than expected, the data are lost in the noise and a great deal of adjustment and recalibration is required. The noise performance of the DATEL digital-to-analog converters is quite superior to that of the tape recorders, and time of setup for data acquisition is an order of magnitude better.

The price that one pays for the versatility and flexibility of such a system is the learning curve that is inherent in its design and implementation. When this penalty is paid, however, the knowledge that is thereby gained gives a far greater measure of control of troublesome situations when they arise, as they inevitably do. The alternative of turning to a manufacturer can be very costly in time and capital. If it were possible to buy a fully engineered system at a competitive price that would perform all of the functions that were needed, one would certainly be money ahead initially, but control of troublesome situations would still be lacking, and sooner or later time would have to be taken to know the system.

If one were to follow to an extreme the above trend of thinking, one would not buy an operating system at all, but rather write it. We have not gone this route, which no doubt has difficulties, but it is possible to buy a minimal operating system and embellish it as one wishes, writing

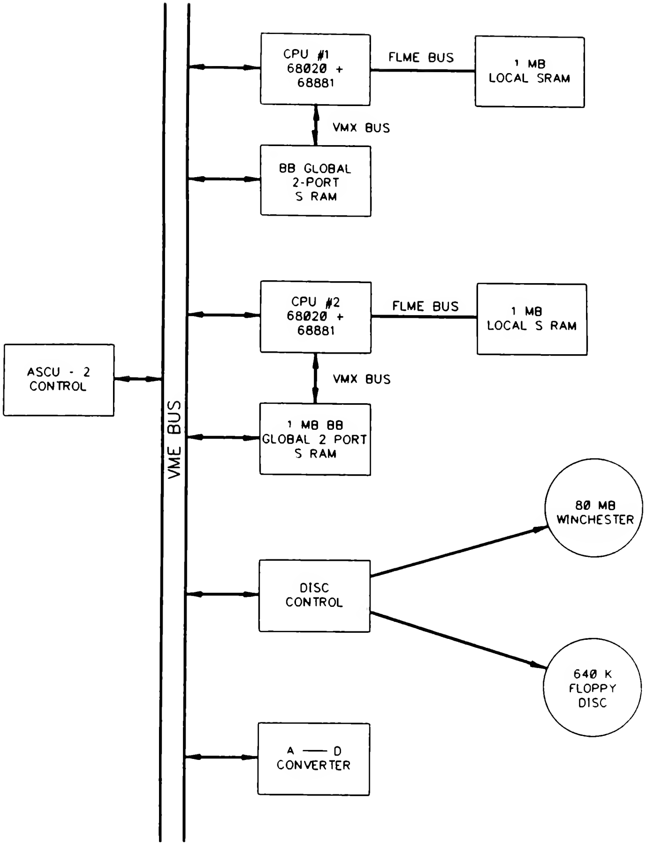


Fig. 2. SYSTEM AS PRESENTLY CONFIGURED

probably in the "C" language. This seems to be a viable alternative, but we cannot speak from experience.

Our graphics are being done on a Tektronix 4207 terminal using the TCS high level software package. The terminal is controlled by this package using calls from FORTRAN, which is not the main high level language of the UNIFLEX operating system (C is). This situation necessitated our purchase of a FORTRAN compiler from ABSOFT, which also allows programmers who prefer that language to use it but restricts the use of real-time operations. The ABSOFT software has been excellent, but it does not contain all the real-time operations that the UNIFLEX system does for C. In light of the above and the facts that TCS does not use the full capabilities of the terminal and that there is a fair amount of work involved in adapting TCS to our system, we probably should not have bought TCS. The time would have been better spent learning the 4207 commands and adapting C to it. This is what we have ended up doing anyway for the sake of real-time capabilities when C is used. Then all of the (in our view) advantages of C become available, most especially the compactness.

Both types of data acquisition, computer and analog tape system, have shown themselves to be

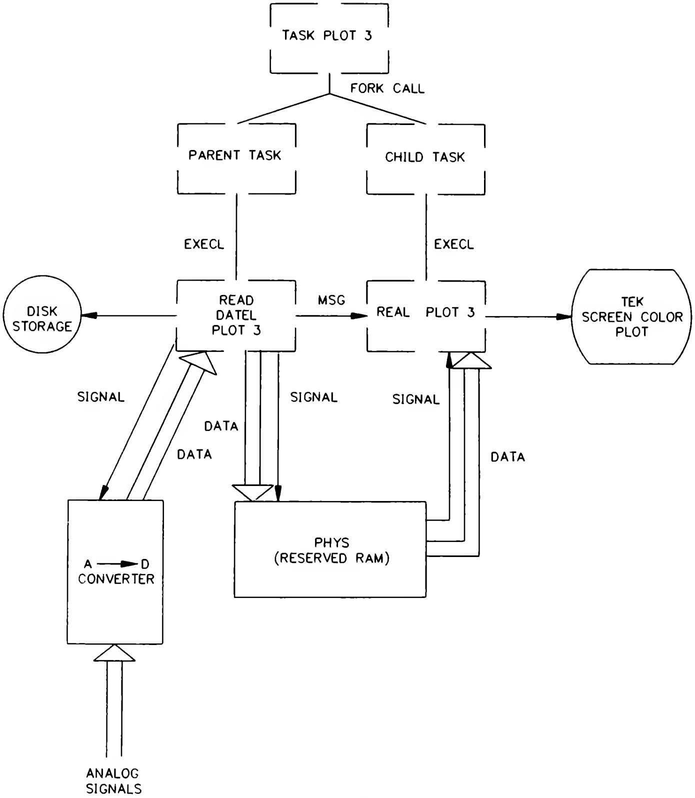


Fig. 3. MULTIPROCESSING FLOW DIAGRAM

bulky to transport, but again the computer and associated hardware are somewhat smaller than the analog tape machine and its complement of peripherals. Custom designed packing cases have been a help in both instances.

IV. Results to Date

We have taken the equipment to several sites for data acquisition and analysis and for demonstration purposes. The reliability of the system has been a source of satisfaction in these situations as has been the ease of setup as compared with analog FM tape methods. Duplicating and safeguarding data with floppy disks has also been easy and fast.

Not only has the equipment been used on site for acquisition and analysis, but it has been used at the Laboratory to analyze data taken by older analog methods, where it has proved to be both fast and easy to operate.

Data have been taken and analyzed at Jacksonville Naval Air Station to determine whether or not MILSTD 1472C has been violated. The data show that the simulators do indeed have frequency components of motion in the range in which the human body has shown most sensitivity, namely 0.2 hertz. It should be noted that, since the simulator has a very limited range of motion compared with that of the aircraft, the motion of the simulator cannot hope to perfectly duplicate that of the aircraft. Consequently some frequencies of motion will be present in the simulator which are largely absent in the aircraft.

V. Future Goals

We are currently scheduled to take the system to Pensacola for analysis of another simulator there in the summer of 1988. We expect to complete the analysis while on site and to deliver the results before we leave the area.

The motion dynamic data acquisition and analysis of the simulator itself are well in hand. We are currently enhancing our analytic capabilities with the addition of transfer function and correlation functions to compare simulator motion with that of the (pilot actuated) controls.

More importantly, perhaps, we need to do some comparison of the visual presentation with the motion of the simulator to examine time correlation between them and to see if the visual imaging is smooth as perceived by the eye.

This last goal will stretch the capabilities of the computer far more than the motion studies because of the volume of visual data per unit time. Special purpose boards have been purchased for data acquisition of frames of video data, but the analysis must be done by the computer. Another CPU or two can be added to the system, if such is needed, and be used effectively. The operating system presently has the capability of handling four CPUs, so the addition of these will present no problem. It is anticipated, however, that doing the visual motion analysis in real time will be difficult, and some algorithms will have to be written to compress the visual data taken.

A METHODOLOGY FOR SIMULATION VALIDATION USING OPTIMAL TIME HISTORY MATCHING

88-4617-CP

Robert A. Hess*
Barbara H. Stanka*
Margaret Bald Purdy††

Systems Control Technology, Inc.
Flight Simulation Group
Lexington Park, MD

ABSTRACT

The technique most widely used for simulation validation consists of matching measured aircraft response with simulation predicted aircraft response. Though this technique is generally considered the best method available for simulation validation, it has shown itself to often be unsuitable for validating aircraft simulations. Specifically, time history matching can be very difficult to perform due to errors in the measured aircraft response, aircraft irregularities, and the inability to isolate the errors caused by inaccuracies in various portions of the simulation model. Presented in this paper is a description of a new technique for simulation validation which alleviates many of the problems associated with time history matching by providing a direct link of modern systems identification with the aircraft simulation environment.

NOMENCLATURE

Symbols

B	measurement noise parameter vector
D	control distribution matrix
F	state matrix
G	control matrix
H	state distribution matrix
J	performance measure
N	number of samples of data
P, Q, R	roll, pitch and yaw rates, respectively, deg/sec
t	time, sec
V	airspeed, ft/sec
W	weighting matrix
x	state vector
y	output state vector

Greek

α	angle-of-attack, deg
β	angle-of-sideslip, deg
Δx_0	process noise parameter vector
δ	control vector
ϕ	roll attitude, deg
θ	pitch attitude, deg

Superscripts

$\cdot$	denotes time derivative of term
$\wedge$	denotes estimated value

MOTIVATION

Modern aircraft simulations have become an indispensable tool for aircraft development and analysis. Such simulations are used in the evaluation of current aircraft systems, and as a tool for familiarizing pilots with the complexities of today's high performance aircraft.

It is important that these simulations be as accurate as possible (i.e. realistically reflect the actual aircraft). Effective, quantitative methods for simulation validation are required so that existing and future simulation models can be verified. However, due to the complicated nature of today's aircraft, validating simulation models has become a difficult task. Furthermore, many of the existing methods for aircraft simulation validation (i.e. frequency matching, trim shots, etc.) have proven to be less than adequate for achieving full flight envelope simulation validation.

The ultimate technique for simulation validation is a comparison of the simulation with the "design basis system", typically an aircraft in flight. In the case of flight dynamics simulation validation, a limited number of parameters about the aircraft are available, usually from telemetered data. Noise and external disturbances lead to the recording of less than perfect data. Furthermore, the test article, while intended to be representative of an entire production run of one aircraft, is actually unique from a flight dynamics viewpoint. Slight aerodynamic irregularities, age and wear on the various mechanical systems, especially turbine engine components and inertial differences, serve to make each test article distinct. These differences complicate the task of comparing simulation predictions of aircraft response to actual measured aircraft response. Measurement errors and aircraft specific irregularities are not accounted for in simulation models. These items cause simulation predictions to rapidly diverge from the measured response. This effectively diminishes the capability of using time history matching for simulation validation.

Presented in this paper is a methodology which accounts for measurement errors and aircraft specific irregularities in time history matching. This methodology uses system identification technology to model and account for the causes of simulation divergence. Hence, a rigorous and practical method for validating aircraft simulations is available.

DISCUSSION

An aircraft simulator predicts aircraft motion by integrating the aircraft equations of motion (state equations) given the initial conditions of the aircraft,

* Engineer, member AIAA
†† Programmer/Analyst

the control inputs into the system (i.e. pilot inputs, air density, aircraft mass and inertial properties) and a model which expresses the functional relationships between the applied forces and moments on the airframe and the states and controls (i.e. aerodynamic, engine and control system models). This process is demonstrated by considering the general equations of motion:

$$\dot{x} = Fx + G\delta \quad ; \; x(t=0) = x_0$$

$$y = Hx + D\delta \qquad [1]$$

From measured flight data, the initial aircraft states (x_0) and control inputs (δ) into the aircraft are recorded. With this information, a time history comparison analysis can be performed. Such a scheme is normally used for a time history matching analysis. A high level representation of this scheme is illustrated in Figure 1.

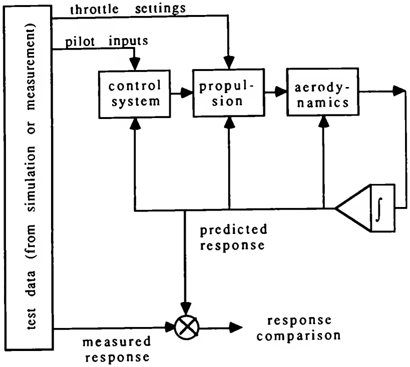


Figure 1 Use of measured aircraft response for simulation time history matching.

Even if the simulation model is accurate, a successful time history matching analysis cannot be guaranteed. Unmodeled or unobserved disturbances acting on the aircraft will produce a prediction which can diverge from the measured aircraft response. To further complicate the issue, errors in measuring the aircraft states and controls in flight can also produce a poor prediction analysis. For example, if data for a high-performance aircraft was collected at an altitude of 25000 ft at a Mach number of 0.6 with the stabilator deflection signal biased by 0.25 deg, then after a ten second simulation period, the pitching moment bias caused by the stabilator bias can integrate out to cause the pitch rate to diverge by a magnitude of ~20 deg/sec! It is evident that these types of errors need to be considered in doing a prediction analysis.

Due to the coupled nature of simulation models, errors in one portion of the simulation can cause other portions of the simulation to provide incorrect responses. An example of this would occur if there was a modeling problem in the control system model. The predicted control surface positions would not match the actual control surface positions. Because of this, the aerodynamic forces and moments which are predicted by the model would be incorrect, and such forces and moments would be integrated out to cause the predicted aircraft response to be quite different from the measured aircraft response.

Based on these types of problems, a tool for simulation validation must be created which can account for process and measurement errors, and also be able to exercise specific portions of a simulation without the modeling inadequacies of other portions of the simulation complicating the validation.

Technique to Account for Data Errors

The first step in accounting for the effects of measurement and process errors in test data on simulation response is to propose a model of the errors. A variety of parameters can be used to model measurement errors and aircraft specific irregularities. This is done by augmenting the general equations of motion

$$\dot{x} = Fx + G\delta + \Delta\dot{x}_0$$

$$y = Hx + D\delta + B \qquad [2]$$

where the $\Delta\dot{x}_0$ term accounts for unmodeled disturbances (process noise) and the B term accounts for measurement errors (measurement noise). This model would generally be formulated such that the process noise terms represent force/moment biases acting on the aircraft, whereas the measurement noise terms would represent biases in the response measurements. Such a modeling technique has been used extensively in the area of systems identification, where these types of errors must be accounted for so that system models can be extracted from test data[1-3]. Some of this work has even addressed the issue of model validation, specifically the modeling of these errors prior to validating aerodynamic models[4]. In order to produce a time history matching analysis between a simulation predicted response and measured aircraft response, the noise terms must be known. As was done in the past, a trial and error approach ('hand-tweaking') was used to get estimates of the noise terms. Such an analysis is generally very slow and inefficient, very possibly taking days just to validate the aerodynamic model against 20 seconds of test data.

One of the authors[5,6] recognized that the noise parameters could be estimated using modern identification techniques. Essentially, the difference between the measured and predicted aircraft response provides information about the nature of the noise terms. Using this information, estimates of the error terms can be found. Such a scheme is the basis of all modern parameter estimation analyses[7,8].

In this analysis, the error parameters are estimated using an output error estimation scheme[9]. Such techniques develop a mathematical construct known as a performance measure (cost function). The performance measured (J) is defined as

$$J(\Delta x_o,B) = \sum_{i=1}^{N} \{y_i - \hat{y}_i(\Delta x_o,B)\}^T W \{y_i - \hat{y}_i(\Delta x_o,B)\}$$ [3]

This measure provides a quantitative assessment of the level of difference between the measured response and the simulation predicted response. Basically, all identification schemes estimate parameters such that the performance measure is minimized. When the performance measure is minimized, the estimated error parameter values are then optimal. This minimization is generally not a trivial task, and many methods[10-12] have been developed to solve this problem. This minimization usually requires an iterative approach to perform the minimization (e.g., parameter estimates are incrementally updated until a minimization occurs.)

In this work, the unknown error terms are estimated in an optimal fashion such that the sum of the squared differences between the predicted output states (V, α, β, P, Q, R, ϕ, θ) and the measured output states are minimized. This process is shown in Figure 2.

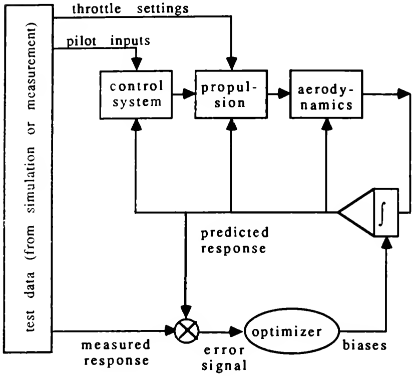


Figure 2 Use of optimization scheme in time history matching analysis.

Assuming that the simulation model is accurate, and that the measurements are not greatly corrupted by errors, the estimated parameters should be relatively small. However, if the estimated parameters are large, then they are not only accounting for process and measurement noise, but also for inaccuracies in the simulation model.

Comparison of the motion prediction of the simulation to the measured aircraft response and an analysis of the estimated parameters are used to develop two criteria which can be used to quantitatively judge the fidelity of the simulation model. A model will have adequate fidelity if :

- o it achieves an adequate prediction of aircraft motion as compared to measured aircraft response
- o its estimated parameters are small in magnitude.

The above criteria can be expressed numerically. This then provides a quantitative measure of the fidelity of a simulation. This method is itself superior to existing methods of simulation validation because it

- o does not need to be executed at trim conditions
- o can analyze a complete maneuver at one time
- o can be automated when implemented into a simulation
- o not only determines if the simulation is valid, but can pinpoint specific regions in the simulation which may be inaccurate (i.e., control system model, propulsion model, longitudinal and/or lateral aerodynamic models, etc.)

Ability to Isolate Specific Portions of a Simulation During Validation

Though the ability to account for process and measurement errors is required in performing a simulation validation analysis, it is also important to be able to exercise specific portions of a simulation so that modeling errors in one portion of the simulation do not affect the response of other portions of the simulation.

Consider the simple problem where the pilot inputs (δ_{stick}), surface position (δ), aircraft accelerations (a), and aircraft states (Y) are known from measurements. This information can be used to construct a simplified system model of the form:

$$\delta(t_1) = f_1(Y(t_1), \delta_{stick}(t_1))$$

$$a(t_1) = f_2(Y(t_1), \delta(t_1))$$

$$Y(t_2) = f_3(Y(t_1), a(t_1))$$ [4]

As is usually performed in validating simulations, the pilot inputs are used as the only inputs in the model, and the model is evaluated for fidelity by comparing the measured and predicted response to one-another. Each of the simulation models for the control system (f_1), aerodynamics and propulsion model (f_2) and integration scheme (f_3) may be in error. Hence, in treating the pilot stick deflections as the only input to the model, it can be very difficult to isolate which portion of the model is in error when the model predicts a response which is divergent from the measured response. However, this problem can be alleviated by using other information about the system in an effort to reduce the divergence. This is simply accomplished by replacing predicted/propagated response from each model in the simulation with actual measured response. For example, if it was of interest to validate an aerodynamic model, it would simplify the validation effort if the measured surface positions were used to drive the aerodynamics instead of the predicted surface positions, since the control surface model may be in error.

IMPLEMENTATION OF VALIDATION SCHEME

Currently, SCT is involved in the development of a new flight dynamics simulation software architecture. This simulation includes a "shell" of generic software, which provides a standard set of operational and analytical commands, a standard set of generic simulation variable names, and well defined initial condition parameter vectors (which can be stored for test repeatability).

The validation scheme discussed above has been implemented into the simulation environment. This implementation has been named SCOPE (Simulation Checking using an Optimal Prediction Evaluation). As with other simulation tools, SCOPE has been set up to run interactively in the simulation to provide simulation engineers and scientists an efficient method for simulation validation.

The implementation of SCOPE into a simulation is presented in Figure 3. The basic components of SCOPE consist of :

- o modules to read user supplied information.
- o modules to reconstruct initial conditions from measurements.
- o modules to interface SCOPE into the simulation so that SCOPE can 'drive' the appropriate simulation modules with measured inputs. For example, the control system model can be driven with measured pilot inputs, whereas the aerodynamic model can be directly driven with measured surface positions.
- o modules to provide the required optimization analysis. This includes modules to calculate residuals, estimate parameter values, determine performance measures, etc.
- o modules to output final time histories and error parameter estimates.

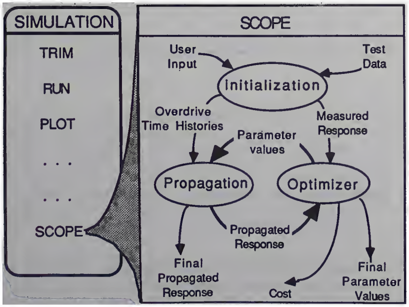


Figure 3 Structure of SCOPE validation module as implemented into a simulation environment.

SCOPE provides the framework for a systematic approach to simulation validation. Initially, the control system and propulsion models can be evaluated (first without the optimizer, and then with the optimizer). Simultaneously, the aerodynamic models (either the longitudinal and/or lateral aerodynamic models) can be evaluated. Furthermore, analysis of the error parameters which are estimated can provide a great deal of information about the deficiencies in each simulation module.

This method also has the advantage of being able to compare different versions of an aircraft simulation. This allows for a definitive comparison of the effective differences between various simulation versions.

Even though this methodology was developed for simulation validation purposes, the identification techniques incorporated into this work could easily be extended to provide a low-level parameter identification tool. This would allow for the identification of not only simple bias parameters, but also of non-linear increments to any specified set of aerodynamic/engine/control parameters.

RESULTS

Presented below are four examples of the use of the SCOPE validation routine. They highlight the basic capabilities of SCOPE in simulation validation. The first example is a demonstration of SCOPE using synthesized flight data, while the second, third and fourth cases illustrate the use of SCOPE with actual test data from two different high performance attack aircraft.

Example 1 - Comparison of Aerodynamic Models from Two Different Simulations of the Same Aircraft

It was of interest to check if the performance of a lateral-directional aerodynamic model in one simulation was equivalent with the performance of a model in a simulation in which SCOPE has been installed. The models to be analyzed were derived from the same data sources, and hence should be equivalent. In this example, data was generated for a 100% deflection of the lateral stick which was held for 2 seconds, and then released. Specifically, it is desired to check whether the force and moment coefficient time histories predicted by each simulation were equivalent. If the models are equivalent, the force and moment coefficients produced by each simulation should be the same for the same control surface motion. Model equivalence was checked by overdriving the surface positions generated from one simulation in the simulation which has SCOPE (Figure 4).

As is seen in Figure 5, both models exactly match lateral-directional response for the specified inputs. This confirms that the models are equivalent.

Example 2 - Comparison of Control System Behavior of a Simulation as Compared to Actual Flight Test Data During a Departure

During an evaluation of a new aircraft configuration, a departure occurred in-flight while the aircraft was performing a series of rolls. In analyzing the control system, measured aircraft state information and pilot inputs were used to drive a simulation model in which SCOPE was installed. This

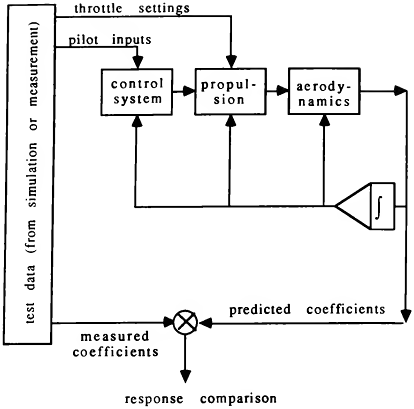


Figure 4 Scheme used to compare different aerodynamic models together using SCOPE.

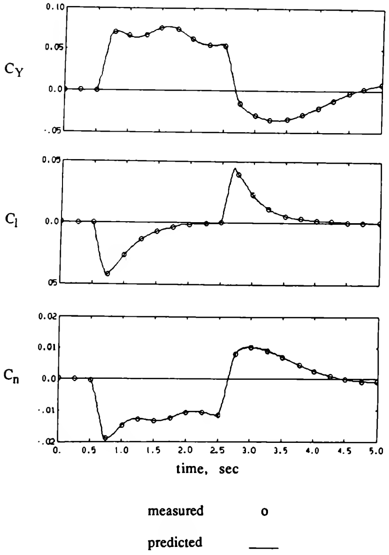


Figure 5 Comparison of response of different lateral-directional aerodynamic models.

scheme is shown in Figure 6. In an effort to minimize control system divergence caused by propagating errors in the aerodynamic and propulsion models, aircraft state information which is nominally propagated in aircraft simulations was overdriven with measured state signals.

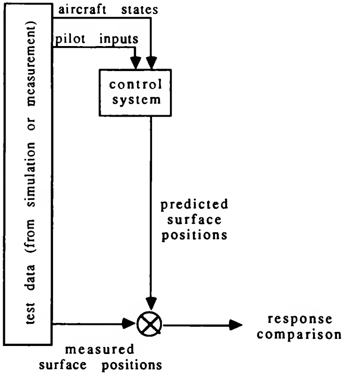

aircraft states
pilot inputs
test data (from simulation or measurement)
control system
predicted surface positions
measured surface positions
response comparison

Figure 6 Use of SCOPE to check the response of a control system during an aircraft departure.

As is seen in Figure 7, the measured and predicted surface motions matched very well. Though there are some small biases (< 1 deg) in some of the measurements of the surface positions, these biases are probably attributable to errors in measuring surface positions, trim offsets caused by aircraft specific irregularities, and possibly structural bending. In general, it appears the control system onboard the aircraft worked as designed.

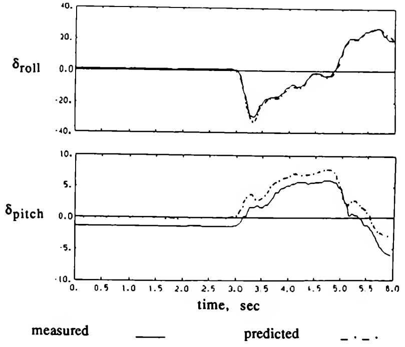


Figure 7 Comparison of measured and predicted control surface positions during an aircraft departure.

Example 3 - Estimation of Aerodynamic Model Errors in a Simulation from Flight Test Data Gathered from Departure Data

The flight test data in the previous example was further analyzed in an effort to understand the aerodynamic behavior of the test aircraft during the departure. It was known from previous parameter identification analyses and analysis of other departures of this aircraft that the longitudinal aerodynamic model was accurate, but that the lateral-directional aerodynamics were of low fidelity.

The first step in this analysis was to check how well the simulation model would predict the forces and moments acting on the aircraft. This was accomplished by overdriving the simulation surfaces and states with measured values, and then comparing the predicted and measured force and moment coefficients to one another (Figure 8).

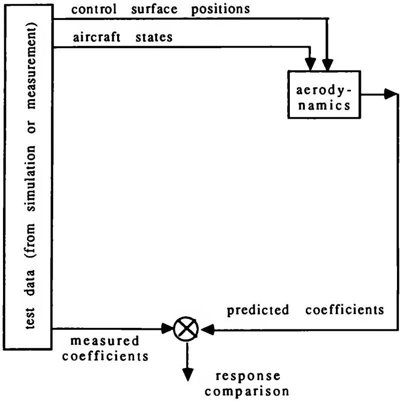


Figure 8 Scheme for comparing simulation and flight test force and moment coefficient time histories.

An example of the results of the force/moment comparison analysis are shown in Figure 9. Illustrated in this figure are plots of the measured and modeled rolling and yawing moment coefficient time histories. The modeled responses showed divergent time history responses as compared to the measured moment coefficient time histories. This indicates that there exists errors in the lateral-directional model.

The time histories of the measured and predicted moment coefficient data were further analyzed to ascertain why there was a modeling problem in the simulation. If it is assumed that there is a relationship between the difference in the aerodynamic coefficient signals and aircraft state and surface information, then a model for this relationship can be formulated. It is assumed that the difference in the coefficients (ΔC_a) can be written as a linear combination of error parameters and aircraft state/control information of the form:

$$\Delta C_a = A\,\mathcal{P} \qquad [5]$$

where the matrix A is composed of the measured aircraft state and control information and has the form:

$$A = \begin{bmatrix} 1 & \alpha(t=t_1) & \beta(t=t1) & \delta(t=t1) & \ldots \\ 1 & \alpha(t=t_2) & \beta(t=t_2) & \delta(t=t_2) & \ldots \\ 1 & \alpha(t=t_3) & \beta(t=t_3) & \delta(t=t_3) & \ldots \\ . & . & . & . & . \\ . & . & . & . & . \\ 1 & \alpha(t=t_n) & \beta(t=t_n) & \delta(t=t_n) & \ldots \end{bmatrix} \qquad [6]$$

and where the vector $\mathcal{P}$ is composed of the error parameters which can represent incremental errors in such quantities as aircraft stability and control parameters or propulsion effects.

Because of the the linear relationship of the error parameters to the coefficient differences, the errors parameters ($\mathcal{P}$) can be estimated in a least-squares sense:

$$\mathcal{P} = [A^T A]^{-1} A^T \Delta C_a \qquad [7]$$

Other statistics about the error parameters can be formulated[13] and used to evaluate the significance of the parameters. Such a scheme is currently being incorporated into SCOPE. With this approach,

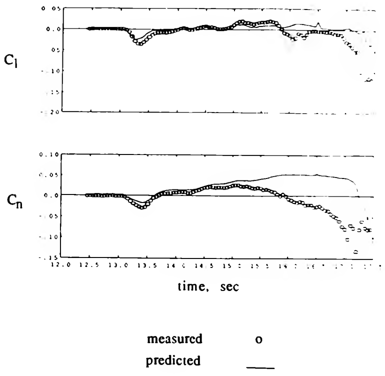


Figure 9 Comparison of measured and predicted moment coefficient time histories for an aircraft during a departure.

increments were estimated for the aerodynamic parameters in the departure data. These increments were consistent with the errors previously determined in the model. When these errors are accounted for, a much improved time history match is accomplished (Figure 10).

Example 4 - Use of Optimization Scheme to Reduce the Divergence of a Simulation and Flight Test Response

As discussed in a previous section, propagation of aircraft states in a simulation may be divergent because the measured inputs into the model may be slightly in error, or because there may be some irregularities in the test article which are not accounted for in the simulation. The use of estimation methodologies in reducing simulation divergence is demonstrated in this example.

It is of interest to evaluate the fidelity of a longitudinal aerodynamic simulation model for a high performance aircraft which was developed using modern parameter identification techniques on flight test data. This analysis was performed by comparing the measured longitudinal response to the longitudinal response predicted by the simulation. This comparison is shown in Figure 11. It is evident in this figure that the predicted response diverges from the modeled response.

The SCOPE optimization scheme was used to estimate increments in the longitudinal forces and moments in order to achieve a better match between measured and predicted aircraft response. The estimated increments are tabulated in Table 1.

INCREMENT	VALUE
ΔF_X	487.98 lb
ΔF_Z	1687.80 lb
ΔT_m	-3087.23 ft-lb

Table 1 Estimated force/moment increments needed to keep predicted response from diverging from measured response.

It is not intuitively obvious whether these increments are large or small. However, knowing the mass of the test aircraft and typical engine performance, the increments represent an uncertainty of ~5% in thrust, ~2% uncertainty in CG location, and ~8% uncertainty in mass. These levels of uncertainty are well within the accuracy of the flight test data measurements. The predicted response of the model with the the use of the optimization scheme is shown in Figure 12. Inspection of this figure indicates the good agreement between the predicted and measured longitudinal aircraft response.

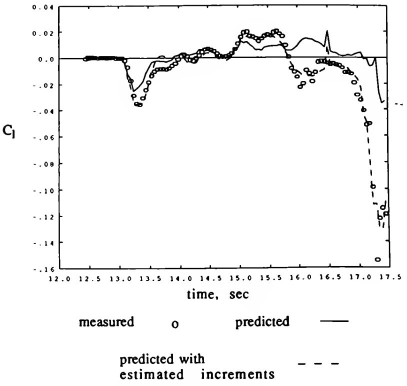


Figure 10 Comparison of measured, modeled and identified moment coefficients for a departure from a SCOPE analysis.

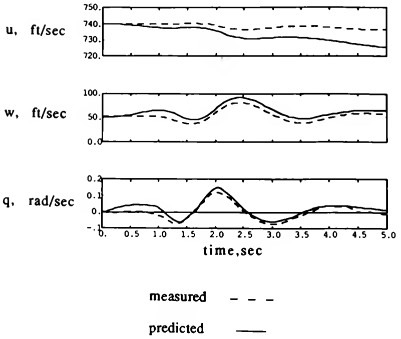


Figure 11 Comparison of predicted and measured aircraft response without use of SCOPE optimizer.

CONCLUSIONS

A technique for the validation of aircraft simulations using an optimal match of measured and simulation predicted time histories has been developed. This method of simulation validation is shown to be rigorous and practical for simulation model fidelity analysis. This technique allows for the validation of either the aerodynamic, engine, or control models separately or together. Furthermore, because this methodology is incorporated into the simulation environment, it can help to automate, and hence simplify, the process of simulation validation.

ACKNOWLEDGEMENTS

The authors of this paper would like to thank Mr. E. Bruce Jackson of the Naval Air Test Center, Patuxent River, MD, and Mr. Laurence C. Anderson of SCICON-SD, Ltd. for their help and guidance. Additional gratitude is extended to Mrs. Theresa Rose of the Naval Air Test Center, Patuxent River, MD, for her cooperation in this work.

REFERENCES

1. Bryant, W., Hodge, W., 'Effects of Flight Instrumentation Errors on the Estimation of Aircraft Stability and Control Derivatives', NASA TN D-7647, April 1974.

2. Bach, R., 'A Variational Technique for Smoothing Flight-Test and Accident Data', AIAA-80-1601, 1980.

3. Keskar, D., Klein, V., 'Determination of Instrumentation Errors from Measured Data using Maximum Likelihood Method', AIAA-80-1602, 1980.

4. Klein, V., Batterson, J., 'Aerodynamic Characteristics of a Fighter Airplane Determined From Flight and Wind Tunnel Data', NASA TP 2483, September 1985.

5. Hess, Robert A., 'Validation of a Fighter Aircraft Simulation Using High Angle-of-Attack Flight Test Data', AIAA-86-2681, October 1986.

6. Hess, Robert A., Anderson, Laurence C., 'Development of an Increased Fidelity F/A-18A Aerodynamic Simulation Model from Flight Test Data Analysis', prepeared under contract no. N00421-85-D-0155 to the NAVAIRTESTCEN, June 1986.

7. Maine, R., Iliff, K., 'Identification of Dynamic Systems', NASA RP 1138, February 1985.

8. Eykhoff, P., System Identification - Parameter and State Estimation, John Wiley and Sons., 1974.

9. Maine, R., Iliff, K., 'Application of Parameter Estimation to Aircraft Stability and Control - The Output Error Approach', NASA RP 1168, June 1986.

10. Klein, V., 'Identification Evaluation Methods', AGARD Lecture Series no. 104, AGARD-LS-104, 1979.

11. Beck, J., Arnold, K., Parameter Estimation in Engineering and Science, John Wiley and Sons., 1977.

12. Bard, Y., Nonlinear Parameter Estimation, Academic Press, 1974.

13. Draper,N., Smith, H., Applied Regresion Analysis, John Wiley and Sons, 1966.

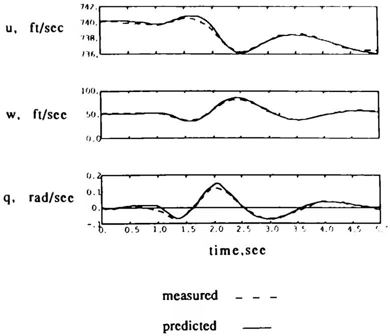


Figure 12 Comparison of measured and predicted longitudinal aircraft response using SCOPE optimization scheme.

88-4618-CP

REAL-TIME SIMULATION - A TOOL FOR DEVELOPMENT AND VERIFICATION

David R. Bloem
System Simulation Staff
System, Analysis, and Simulation Department

Robert Naigus
Systems Engineer
Commercial Sŷstems Department

Smiths Industries
SLI Avionics System Corporation
4141 Eastern Avenue
Grand Rapids, Michigan 49508

June 16, 1988

Abstract

Smiths Industries SLI Avionic System Corporation (SLIASC) has recently developed a baseline real-time laboratory simulation tool for the purposes of development and verification of a wide variety of avionics systems ranging from a Flight Management System to a Navigation Attack System. This laboratory tool is used for all aspects of product development including, system/software development, full system verification and validation, on-site flight test support, and field customer training support. This paper discusses a wide range of topics from the laboratory design methodology, and its associated configuration, to the specific design features of the real-time simulation. It presents an in-depth look at the advantages of developing common laboratory software and hardware, and applies it to the different stages of product development. Finally, it takes a look at engineering simulation analysis and test facilities.

Introduction

SLI Avionic Systems Corporation (SLIASC) based in Grand Rapids, Michigan is part of the Aerospace and Defense group of Smiths Industries. The corporation develops a wide range of commercial and military avionics systems, as well as military reference systems for the primary purpose of providing vehicle management. These systems include a Flight Management Computer System (FMCS) on the B737 aircraft, a Navigation Attack System (NAS) on the A-4 aircraft, a Self-Contained Navigation System (SCNS) on the C-130 aircraft, and advanced conceptional studies of military flight management. This paper describes the real-time simulation tool used for development and verification of the products mentioned above from the simulation development point-of-view, as well as from the user (application) point-of-view.

Simulation Development Approach and Methodology

In the process of developing and testing these products, it became necessary to develop a baseline real-time simulation tool that could be applied to a variety of programs with little or no modification required. The development approach of this simulation effort is to identify the common requirements among different products, and apply them to the different phases of product development, in order to build a baseline real-time simulation tool that minimizes the cost of product development and testing.

The main thrust is to identify the functions that are required by the majority of the programs. These functions become the major building blocks of the baseline tool. In this effort, some of those functions include: the executive scheduling function, flight environment modeling function, main I/O function, operator/display function, data recorder function, and the software development function. Those functions are tied together by a global section and applied to the many different phases of product development. Those phases include: software development and verification, hardware/software integration test, system verification and validation, on-site flight test analysis, and on-site customer system training and support.

The commonality of requirements among the different products is very difficult to achieve, but has three (3) major advantages. First, it reduces the man hours required for developing the tool for each product. Second, the tool is available earlier in the development cycle, and third, the common design structure and operator interface increases the flexibility of assigning manpower to support the testing of the product from the service point-of-view as well as the user point-of-view.

Laboratory I/O Emulation Hardware

The laboratory test station hardware components consist of a variety of devices required to emulate thè actual product's installation into the vehicle, along with providing the additional capability to verify that the product is operating correctly. There are four (4) different hardware devices that are part of the basic laboratory test station I/O configuration. This configuration handles approximately eight-five (85) percent of the I/O emulation. Some projects require special hardware in addition to the hardware described in this section. Since the configuration is defined via an input file, not all the hardware is required at every test station. This greatly reduces the cost of the laboratory test station, since not all stations are performing the same function as well as communicating to the same product. A diagram is presented in Figure 1 illustrating the laboratory test station hardware components described below.

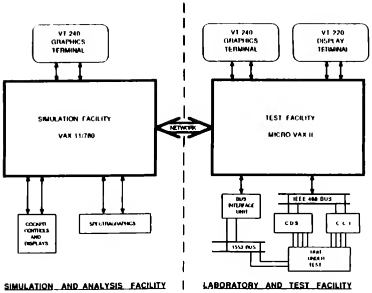


Figure 1

The laboratory test station host computer is a Digital Equipment Corporation (DEC) VAX. The type of VAX is a function of many different factors ranging from physical site requirements to computational power requirements needed to perform the modeling and I/O emulation. The execution of the baseline tool is independent of the test station host provided that there is enough computational ability to support the configuration desired. The test station host VAX has several I/O devices connected to it using one (1) of three (3) possible interfaces for each device. They include: a IEEE-488 bus, a RS-422 serial I/O port, or direct memory access (DMA).

The first I/O device is made up of Colorado Data Systems (CDS) 53A equipment, which provides a standard I/O interface for which many functions can be emulated. The functions included depend on the configuration required to emulate the product's installation. They include: analog input/output relay discrete cards, ARINC 429 transmitter/receiver cards, analog/digital converter cards, digital/analog converter cards, digital/synchro converter cards, and synchro/digital converter cards. This equipment communicates to the test station host via the IEEE-488 bus.

The second I/O device is call the "Computer Control Interface Unit" (CCU). This equipment provides an interface to the Units(s) Under Test (UUT) processor(s). This is the hardware device that performs the software development function of the product's development cycle. It's function is to control the UUT processor execution, and capture data for the purposes of debugging the Operational Flight Program (OFP) software. It consists of a Z8000 processor with 32K words of EPROM and blocks of 16K words of RAM. A more detailed discussion follows in the FMC application section.

The third type of I/O device is call the "Bus Interface Unit" (BIU). This hardware emulates the MIL-STD 1553 data bus traffic used in most military applications. The BIU communicates directly to the test station host via direct memory access (DMA). The application is the emulation of multiple sensor subsystems residing on the 1553 data bus (Inertial Navigation Unit, Digital Flight Control Computer, etc.).

The fourth I/O device is a special high-speed Analog/Digital (A/D) and Digital/Analog (D/A) converter card, that is connected directly to the test station host via direct memory access. The A/D conversion is used for controlling the vehicle via the control inputs from the cockpit. The D/A conversion is used for strip chart recording and/or *x-y* plotting of rapidly changing data.

Laboratory Display Hardware

The laboratory display hardware components consists of three (3) major items: a graphics display system, cockpit instrumentation, and a DEC VT-240 graphics display terminal.

The graphics display consists of a symbol generator unit with a 19 inch high resolution display monitor. The graphics system consist of 1 mega byte of display list memory with the capability of displaying up to 256 colors simultaneously. This gives the laboratory the capability of rapid prototyping of a variety of displays ranging from control display pages to cockpit instrumentation.

The cockpit consists of a set of instruments to display the current state of the vehicle along with three (3) devices to control the vehicle. The controls consist of a control stick, rudder pedals, and a throttle. The set of instruments includes: an attitude direction indicator (ADI), horizontal situation indicator (HSI), airspeed indicator, altimeter, and a vertical speed indicator.

The cockpit is used for several development and testing functions, which include man-in-the-loop workload evaluations, testing of the UUT outputs that control the vehicle, and for demonstrations and training of the entire avionic system in a realistic product environment.

Laboratory Software Components

The laboratory test station software components consists of a variety of processes resident in the test station host VAX. They range from a process that models the product's environment to a process that records data in real-time. There are seven (7) possible processes depending upon the mode of operation and the associated hardware configuration of the test station and/or facility. The first three(3) processes described in this section define the minimum software configuration. The remaining four (4) processes are all optional depending upon hardware configuration as well as product development cycle. A diagram illustrating the possible software configurations is presented in Figure 2.

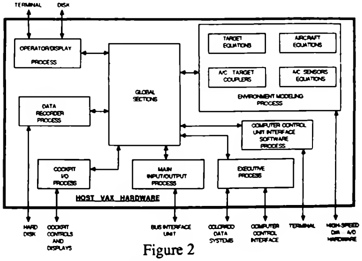


Figure 2

LABORATORY SOFTWARE COMPONENTS

The executive process has two (2) main functions to perform. The first is to provide a timing base for the entire simulation system in order to schedule other processes by setting the appropriate event to trigger execution. The second function, optional depending on the laboratory configuration, is to perform I/O via the IEEE-488 bus to/from the CDS equipment and the CCU equipment described in the hardware section.

The Computer Control Interface Software (CCUIS) process is a non real-time process that executes upon entering a command via the terminal. It performs the software development function of the product's development cycle. The CCUIS process via the CCU equipment has the capability of loading the UUT with Operational Flight Program (OFP) software from a VAX. It also has the ability to examine a given address in UUT memory. This is very important for almost all phases of product development, since it tells the engineer if the UUT is operating correctly.

The operator/display process has a variety of functions and is the operator interface to the simulation system. The most important part of this process is the capability of displaying all the data in the global section in the appropriate engineering system units. This process also manages the input files and performs the initialization (start/up) of the laboratory simulation run. The start/up procedures reads in the appropriate data to determine the configuration and installs the proper processes required to support that configuration. This process is also responsible for providing commands to invoke certain real-time simulation design features requested by the user. These features are discussed in the section called "Real-Time Simulation Design Features."

The environment modeling is an optional process that is responsible for providing real-time flight environment data to the UUT at its execution update rate. This process is used as part of the full system verification and validation of the product's development cycle, as well as hardware/software integration test. The data generated in this process is used by the main I/O process, data recorder process, cockpit I/O process, and the operator/display process. This communication is accomplished by a global section resident in the test station host VAX. This process is linked by modules that represent certain models of the product's environment. They include modeling the aircraft flight, flight couplers, air-to-ground and air-to-air targets, and sensor subsystems.

The main I/O process is an optional one, based on the hardware configuration associated with the CDS equipment in the laboratory test facility. This process performs a variety of tasks which include preparation of data for communication to the executive process or to the actual interface, depending on the type of signal being processed. The process is also responsible for determining when the signal is to be updated and on what frame the update should take place. This data, together with the configuration, is specified by the user in a standard input file. Each configuration or user has its own file, which enables multiple users to use the main I/O process without modification. This greatly reduces the cost of developing simulation test laboratory I/O software.

The data recorder is a operator initiated process that saves 256 words per frame of global section data at a given delta time. This process writes unformatted data out to a hard disk associated with the test station host VAX. This data is read in by a post-processor program and produces a readable format for data analysis.

The cockpit I/O process is optional and is only required if the cockpit is in the hardware configuration. This process interfaces with the man-in-the-loop cockpit located in the simulation and analysis facility. It contains three(3) functions necessary to drive the cockpit environment as follows: First, scaling of parameters necessary to drive analog instruments along with synchro instruments. Second, it is responsible for accepting the control stick inputs and converting them into the appropriate engineering system units for the environment modeling process. Third, it is responsible for driving the graphics system displays with vehicle data.

Engineering Simulation, Analysis and Test Facilities

The engineering simulation, analysis and test facility at SLIASC, shown in Figure 1, is divided up into two (2) facilities: a simulation and analysis facility and the laboratory test facility. The simulation and analysis facility consists of four (4) CPUs with approximately 3.0 MIPS of capacity and services approximately 100 engineers. The laboratory test facility consists of many micro-VAX-II as the test station host. This facility is responsible for support of the product on-site during flight test and field customer training. Both facilities are primarily made up of DEC computers ranging from VAX 11/780 to micro-VAX-II depending upon site requirements.

Flight Management Computer (FMC) Application

The real-time simulation has many features to assist the system/software personnel develop and test the Unit Under Test (UUT) hardware and its associated software. These features will now be discussed in the context of a specific application - the commercial flight management computer (FMC) for the B737-300/400/500 aircraft. The first major stage of product development to make use of the real-time simulation is software development and checkout. Software engineers can easily check the ramifications of a software change by modifying the FMC code via the CCU and then flying a simulated flight scenario on the laboratory test facility.

Software development and verification as well as system verification and validation is done primarily at the laboratory test facility. Test scenarios designed to exercise the full capability of the FMC are flown using the real-time simulation. These tests consist of formal, or procedural scenarios and also informal (ad hoc) scenarios.

Problem reports received from the airframe manufacturer are investigated at the laboratory test facility. If the problem can be duplicated the fix is usually near at hand. Problem fixes are then verified using the real-time simulation to repeat the problem scenario such that the problem is no longer observed. These techniques can be extremely cost and time effective when responding to airline customer requests, compared to sending a company field representative to the airline directly.

The real-time simulation is used for training various personnel in the latest operational aspects of the product. These personnel include company field representatives, airline engineering and maintenance staff, and pilots. Demonstrations are also given to representatives of industry and potential airline customers, thus the real-time simulation is a valuable tool for sales support.

The Flight Management Test Facility

Figure 3 shows the system components of the laboratory test facility used in development of the commercial Flight Management Computer. Figure 4 shows an actual illustration of the facility.

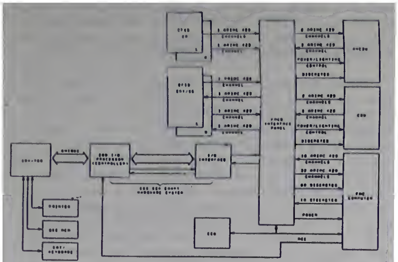


Figure 3 - FMC Lab Test Facility

Figure 4 - FMC Lab Illustration

The Unit Under Test (UUT) is the Flight Management computer, together with its operator interfaces, the Control and Display Units (CDU/ANCDU). FMC capabilities to be tested include: flight planning, performance management, lateral guidance and steering, vertical guidance and steering, and navigation.

Graphic outputs from the FMC are transmitted to the Electronic Flight Instrument System (EFIS). This system includes an Electronic Attitude Direction Indicator (EADI), an Electronic Horizontal Situation Indicator (EHSI), and a separate control panel for adjusting scale and data display on the CRT displays.

The host computer for this facility is a DEC VAX 11/780; herein resides the real-time simulation. Peripheral devices include CRT keyboards for interface, type drives, disk drives, and printers.

The CDS 53A "Smart" Hardware System monitors and controls and flow of ARINC 429 digital data between the FMC and the real-time simulation. This system represents the normal flow of input data to the FMC from other avionics systems on board the aircraft, for example, a position input from the Inertial Reference System (IRS). Likewise the FMC outputs to other systems are represented, for example, a roll steering command to the autopilot computer. The CDS 53A system is composed of a Z80 I/O processor, and I/O interface, ARINC 429 transmitters/receivers, and a discrete (8 bit parallel) interface.

The Unibus Microcontroller (UMC) is a "customized" interface between the VAX 11/780 and the CDU system which allows the FMC to "talk" to the real-time simulation.

The Computer Control Interface Unit (CCU) is composed of a CCU interface card in the UUT, CCU firmware, CCU software, and a control terminal. The CCU card is removed for final FMC testing and normal FMC operation. CCU capabilities include: loading Operational Flight Programs (OFP) into the FMC, stopping the FMC processors, searching FMC memory, changing FMC memory, and restarting the FMC processors. Using the tool a change can be made to the FMC OFP, and the effects investigated immediately via the real-time simulation.

The Interface Panel is a project dependent structure which houses most of the components of the facility. Other components include: semiconductor (RAM) memory panels for the FMC processors. These units are connected to memory interface cards in the FMC; they provide the means for flexibility during the software development phase of product development (load memory, change memory, etc.). The memory interface cards in the FMC are replaced by Erasable Programmable Read-only Memory (EPROM) cards for final system test and actual flight operation.

The FMC Navigation Data Base (NDB) may be loaded as in normal airline operation, i.e., tape cartridge loaded via a data loader, or it may be loaded directly from stored files in the VAX 11/780. The latter method becomes more desirable as the number of customer airline navigation data bases increases.

The Real-Time Simulation Design Features

Upon starting a simulation session, the operator is prompted, via a menu display, to initialize the real-time simulation. There are four main menus from which to choose. If no new initialization data is required, the menus can be skipped and the simulation started. In this case initialization values from the previous simulation run will apply. The first menu contains real world condition data such as position, altitude, heading, weight, fuel, etc. These variables can be set to any desired values within the data entry ranges. The second menu contains analog or discrete data that is normally input to the FMC, for example, a discrete for wing anti-ice. This discrete can be set to "ON" or "OFF" representing the position of the actual flight deck switch that controls this feature. The third menu contains program pin data that is normally grounded "ON" or "OFF" through aircraft wiring. These program pins can be used to activate certain features in the flight software that are desired only by some airline customers. The fourth menu contains error schedules for the inertial reference system (IRS) models. Thus, the operator can introduce, for example, a gyro bias or drift rate into one of the two IRS that will take effect later in the simulated flight. Errors may also be introduced in other simulated systems, such as the radio navigation data, i.e., VOR bias.

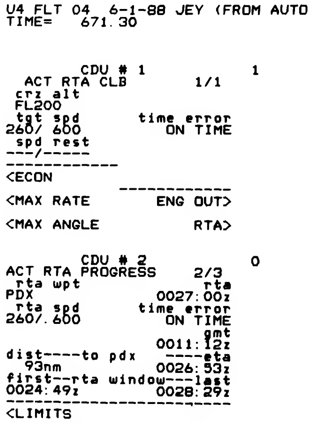
Simulation time is advanced in one of two manners: "clock," wherein the simulation runs on its own internal clock, or "breakpoint," wherein the simulation time advances whenever a particular program address in the FMC is reached. The latter method allows the operator to start and stop the FMC from the simulation.

Simulation updating can be multiplied by an advance factor which affects the propagation of time, longitude, latitude, altitude, and fuel consumption. The factor ranges from zero to 99. Aircraft speed is not affected. Thus a factor of one represents normal speed while a factor of ten will propagate the aircraft at ten times normal speed. This feature is especially useful in reducing actual wall clock time required to get to a specific point in a flight scenario. A factor of zero will keep the aircraft stationary in flight without stopping the FMC processors. In this way all output information displayed on the CDUs can be observed at a "frozen point in the sky."

The operator can, if desired, save all CDU key presses and simulator commands during a simulated flight. In this operation the electronic codes produced by CDU keypresses are saved versus time in the host VAX. Likewise, the VAX commands from the simulation terminal are saved. The run can then be automatically rerun without the operator being present. The file of keystrokes can be edited prior to rerun. This feature is extremely useful in attempting to recreate problem scenarios and it effectively increases computer resource utilization (long, low risk runs can be made during second or third shift without engineers present).

Digital input to the FMC is structured according to the standard of ARINC 429. Using the bit manipulation feature the operator can select any digital input work and toggle any of the 32 bits therein, then send that data to the FMC. Thus it is very easy to change the value of the input data and monitor the FMC's reaction. Data status can also be changed among its various states: normal, fail, test, and "no computed data "(NCD). In this way the operator can easily simulate sensor failures.

During simulated flight and without stopping the FMC processors, the operator can save certain sets of internal FMC data on demand via a single keypress on the control terminal. Examples of this data are the lateral reference path buffer produced by the lateral guidance function and the complete flight plan buffer produced by a combination of the lateral guidance, vertical guidance, performance, and path prediction functions. Data sent from the FMC to the CDU for display is intercepted and saved in a similar manner. An example of the CDU data snap is shown in Figure 5. This data is saved on disk or tape and can be displayed or printed later for analysis.



```
U4 FLT 04  6-1-88 JEY (FROM AUTO
TIME=    671.30



          CDU # 1              1
  ACT RTA CLB         1/1
 crz alt
 FL200
 tgt spd        time error
260/ 600           ON TIME
 spd rest
---/-----
------------
<ECON
            ------------
<MAX RATE         ENG OUT>

<MAX ANGLE            RTA>


          CDU # 2              0
ACT RTA PROGRESS      2/3
 rta wpt                  rta
PDX                  0027:00z
 rta spd         time error
260/.600            ON TIME
                          gmt
                     0011:12z
dist----to pdx      ----eta
  93nm               0026:53z
first--rta window---last
0024:49z             0028:29z
-------------------------
<LIMITS
```

Figure 5 - CDU Data Snap

The operator can also select parameters of interest to be recorded continuously throughout the simulated flight. This data consists of all internal (local variables) FMC parameters and all simulation variables. Again the data, once stored, can be analyzed later. For example, a time history plot of FMC position and simulator position can be analyzed to determine FMC navigation accuracy results.

During the simulation run the operator can insert real-time commands that control simulation models in the following categories: aircraft characteristics (examples: weight, inertia), aircraft subsystems (examples: engines, other avionic systems like the IRS), aircraft flight deck controls (examples: various mode selections of the autopilot), and environment (examples: altitude, heading). Thus, a typical flight scenario might include one or more of the following commands: manually tune the radios, fail the aircraft sensors, adjust the winds, set Mode Control Panel (MCP) inputs, perform baroset, etc. For realism and timing considerations, MCP commands actuated by button presses on the aircraft are also represented by single button presses on the simulator control terminal.

Data collected and recorded for later processing (discussed previously) is also displayed continuously. This data includes: simulation parameters, external input and output FMC data, and internal FMC data. The data is automatically refreshed at a twenty (20) hertz rate. Graphics terminals at the laboratory facility furnish plots of current aircraft position, current FMC flight plan, and current FMC lateral reference path. Aircraft position is updated automatically and the flight plan and lateral reference path are updated on operator demand.

A Typical Flight Scenario

Figure 6 shows a typical flight scenario including climb, cruise, and descent flight phases and flight around lateral navigation waypoints. Table 1 shows typical operator interface commands with the real-time simulation during the simulated flight scenario.

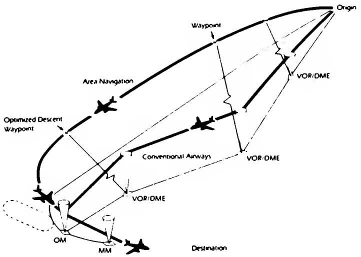


Figure 6 - Typical Flight Scenario

Table 1: Flight Scenario Itinerary

ITINERARY	OPERATOR ACTION
Phase 1 - Preflight	
	Initialize the simulation - position, heading, altitude, weight, fuel, discretes, program pins, sensor errors, etc.
	Start the simulation.
On the ground at KBFI	Initialize the FMC - position, flight plan, performance data, OAT.
	Enter sim commands - MCP altitude, MCP speed Takeoff - equivalent to throttles forward and press TOGA
Phase 2 - Takeoff	
When altitude reaches 400 ft. above airport	Engage LNAV.
	Engage VNAV.
Phase 3 - Climb	
Top of climb (TOC) passage	
Phase 4 - Cruise	
	auto tune, manual tune.
Top of descent	Dial MCP altitude down.
Phase 5 - Descent	
Phase 6 - Approach	
Prior to localizer and glide slope intercept	Tune the localizer, arm the localizer, extend flaps, etc.
KPDX arrival	

Conclusion

Smiths Industries SLI Avionic Systems Corporation has developed a baseline real-time laboratory simulation tool for use in development of its line of avionics products. This tool is chiefly designed to support software development and testing, however it has also been effectively used in support of flight test, field customer service, training, and sales/marketing.

References

737-300 Dynamic Simulator for the Flight Management Computer System, PYV1208, Smiths Industries, SLI Avionics System Corp., Grand Rapids, Michigan February 1, 1985

Colorado Data Systems, 53A Smart Hardware Operating Manual, Colorado Data Systems, 3301 West Hampden Avenue, Englewood, Colorado 80110 Volumes I and II, pages 257-1, 257-11, 420-1, and 423-11

Smiths Industries SLI Avionic System Corp., Software Life Cycle Requirements, Interoperational Procedures 4025, Volume II, pages 1-17, Smiths Industries SLI Avionic System Corp., 4141 Eastern Avenue, Grand Rapids, Michigan August 1986

D. J. Hatley and I. A. Pirbhai, Strategies for Real-Time System Specification, Dorsett House Publishing, 353 West 12th Street, New York, New York 10014 pages 27-31, 1987

AIAA Paper No. 88-4619-CP, AIAA Flight Simulation Technologies Conference
7-9 September 1988, Atlanta, GA

SIMULATOR TRANSPORT DELAY MEASUREMENT USING STEADY-STATE TECHNIQUES

William V. Johnson
Matthew S. Middendorf

Systems Research Laboratories, Inc.
A Division of Arvin/Calspan
Dayton, OH 45440

ABSTRACT

This paper describes a flight simulator transport delay measurement technique along with detailed apparatus descriptions and application considerations. The frequency domain method described was used to measure the delay in a flight simulator used for research investigating temporal fidelity effects on human performance. The transport delay is differentiated from the total delay in the system. Further, time delay contributions from each part of the simulation are described.

INTRODUCTION

Excessive transport delay in a flight simulator can have deleterious effects on pilot performance and training.[1] It is, therefore, important to understand and be able to quantize the transport delay in a flight simulator. Simulator transport delay is defined as the time delay between pilot input and pilot cueing solely due to simulator implementation. Delay due to the dynamics inherent in the real aircraft are not part of the transport delay. These inherent delays must also be quantified so that the transport delay can be computed.

Much of the diversity in quantifying transport delay comes about because of the myriad of simulation implementation strategies. For example, a Computer Generated Image (CGI) that updates at varying rates causes a range of possible transport delay times. Multi-computer simulations that are not synchronized also provide a range of measurable delay. Non-linear aerodynamic models present complications in the measurement of delay. In addition, the different techniques used to measure delay can yield different answers.[2]

Time-domain delay measurement techniques involve measuring the system response to a step input. Methods that rely on measurement of transients can give misleading results when applied to all digital simulations. For instance, a step response measurement on a CGI could miss the delay due to holding effects of a multi-refresh image which become apparent only in steady state excitation. Transient methods also suffer from the intrinsic dependence on the definition of the stimulus and the response.[2] Frequency domain measurement techniques, on the other hand, provide more consistent results.

Research efforts at the USAF Armstrong Aerospace Medical Research Laboratory concerning simulator temporal fidelity require simulators with accurate control of delays. Further, these delays are required to be programmable. Verification of delay is paramount in such research. In designing flight simulators specifically to support research into simulator delay effects, the exacting nature of frequency domain delay measurement is proving invaluable in verifying and accurately quantifying simulation delay.

SIMULATION SETUP

The simulation represented a fighter-type aircraft flying at a constant airspeed. The simulation pilots task was to maintain a constant heading and altitude in the presence of turbulence. Pilot commands were in the form of roll and pitch rate inputs to an isometric (force) stick.

For delay measurement analysis purposes, the simulation was logically divided into the following parts: the simulation computer, graphics computer, and visual display. The simulation computer's job can be further subdivided into computations and input/output (I/O) operations.

A PDP 11/60 digital computer running the RSX-11M operating system was used for the real-time simulation computer. The graphics computer was a Silicon Graphics Inc. IRIS 3020. It is a raster system capable of 60 Hz non-interlaced high resolution color video. The visual display was presented on a 19 inch diagonal raster scan monitor. Graphics were presented with a resolution of 1024 x 768 pixels that refreshed at 60 Hz, non-interlaced.

SIMULATION OPERATION

The order of operations within each simulation frame was: 1) read all inputs, 2) update all outputs, 3) calculate new model states, 4) send aircraft state deltas to the graphics computer, and 5) store simulation states. Analog outputs were changed according to states calculated in the previous frame so transitions became independent of calculation time. Graphics data was sent to the graphics computer at the end of each model solution and was buffered until the start of a graphics frame.

Simulation computer frames were synchronized with the graphics computer frames by detecting the vertical sync of the graphics computer RGB drive signal. Complete synchrony of the simulation was achieved in this way.

The dynamics were implemented using a simple transfer function model. Transfer functions were implemented using a bilinear transformation to derive digital filters. Integrations were performed using the trapezoidal method. Gravity coupling of roll to body yaw rate was used to facilitate coordinated turns. Body translation velocities were calculated from angle of attack and speed. Velocity vectors were then transformed from the body to the world coordinate system.

An out-the-window display was generated using aircraft attitude and position. The display consisted of a flat green earth overlayed with a black grid and covered with a blue sky. The surface was randomly textured with rectangular buildings of different sizes. The display was double buffered so that graphics updates were synchronized with vertical retrace times. The information on the display updated at 15 Hz corresponding to 4 screen refreshes.

Aircraft states were sent to the graphics computer every simulation frame in the form of differences in (deltas of) position and attitude. A delay queue implemented in the simulation software was used to provide programmable transport delay additions in whole frame increments for research purposes. The delay queue stored aircraft state updates for the programmed number of simulation frames before sending them to the IRIS.

The data packets were buffered on the graphics computer. The deltas from all available packets were integrated in the graphics computer at the beginning of each graphics calculation to maintain current aircraft states. The integration produced states current to the beginning of the previous simulation frame. A communications delay of one frame time was expected at this point.

MEASUREMENT APPARATUS

The primary measurement instrument was a Bafco Inc. Frequency Response Analyzer (FRA), model 916. The FRA provided a stimulus (output) signal at a single frequency. Two auto-ranging input channels were available. The FRA calculates the phase difference and amplitude ratios from the two inputs. The signal parameters are derived from an analog implementation of the Fourier integral on the two channels. The integration

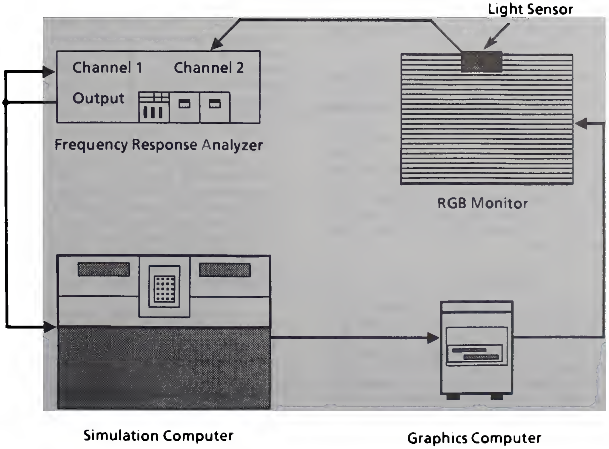


Figure 1 Delay Measurement Setup

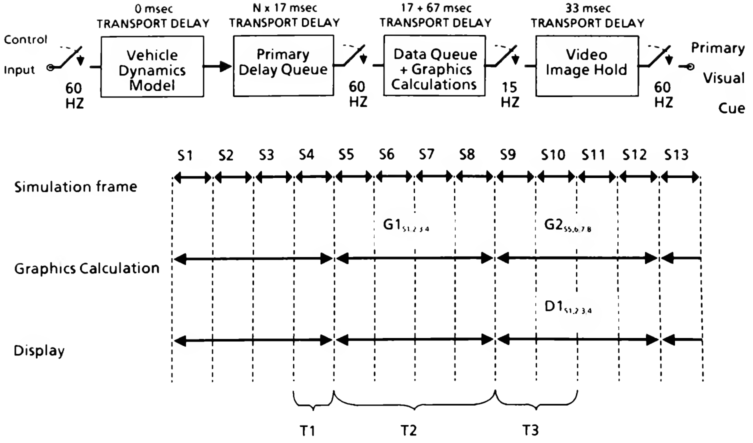


Figure 2 Simulation Timing Diagram

period is an integer multiple of the output frequency providing excellent harmonic rejection. This harmonic rejection capability meant that fundamental response could be accurately read through a non-linear system response.

Simulation response was obtained by sensing light from the display monitor. The light sensor was based on a photo-transistor capable of sensing visible light from the display monitor. It drove a comparator circuit with an adjustable threshold. The digital output of the comparator was connected to a one-shot circuit that was tuned for a 16.67 mS pulse time. This hold time corresponded to the display refresh time. The sensor focused on a small spot on the screen and provided a binary representation of the screen luminance at that point.

METHOD

The phase shift through the simulation, from stick input to display device, was measured at a fixed frequency. The phase shift due to the dynamics of the vehicle were subtracted from the total measured phase shift. The difference represents the phase shift due to the transport delay at the input frequency. The transport delay from phase measurement calculation is represented in Equation (1).

Transport Delay(Seconds) = (1)
(Total Measured Phase(Deg) -
Vehicle Dynamics Phase(Deg))
* (Period of Input Frequency(seconds) / 360)

The simulation was driven with the test signal by patching the output of the FRA in place of the force stick. The pitch axis was selected because body pitch provides a stationary transfer function, with roll equal to zero, to the world referenced pitch available at the graphics computer. The amplitude of the driving signal must be great enough to overcome the control breakout level. Since the aeromodel was rate controlled, any offset in the driving signal was integrated causing drift in the world pitch position. This problem with the technique can be managed if the driving frequencies are high enough so that the measurement intervals are short and do not allow drift errors to become too large. At the other end of the spectrum, frequencies must be less than half the graphics update rate for proper sampling by the sensor. In addition, higher frequencies are attenuated by the dynamics causing the signal to drop into the noise level.

The phase shift due to the dynamics must be known so it can be subtracted out. The simulation software output pitch to an analog channel. This signal was fed into the FRA to obtain the phase of the pitch dynamics. In this measurement, phase due to an additional frame of delay because of the order of operations of the simulation and a one half frame delay due to the ZOH effect of signal reconstruction was accounted for.

The monitor was modeled as a zero order hold (ZOH) device. A ZOH function introduces a time delay of one half its update interval. ' While the monitor refreshed at 60 Hz, the graphics updated at only 15 Hz. The update rate (verses the refresh rate) was found to be the effective rate at which the ZOH was applied. It should be noted that this model assumes that the new graphics information is available at some instant and then is held. In actuality, the new information is presented over the first monitor refresh field of the update and held during subsequent refreshes.

To measure the delay through the visual display, the graphics software was modified. The intensity of the display was set high for a pitch greater than or equal to zero and low for a pitch less than zero. This simple modification did not affect the update time of the graphics computer. The equipment was set up as shown in Fig. 1. The sensor was positioned at the top of the screen to represent a data hold coincident with the start of the display scan. The sensor comparator threshold was adjusted to switch between the two luminance values. The conditioned output of the sensor was connected to the FRA to obtain phase measurements. The binary representation of the signal produced significant noise in the phase readings. To reduce measurement noise, the FRA has an extended integration period setting, and, in addition, four readings are taken and averaged at each frequency. The resulting data table is shown in Table 1.

RESULTS

The transport delay components of the visual path are shown in Figure 2. There is a sample-time to graphics-calculation delay, T1, extending from the sampling of the stick until a graphics calculations starts that uses this information. This delay would include any delay due to the delay queue. The dynamics introduce their phase at this point which, again, is not included. Next, there is a graphics calculation delay, T2. This delay is an integer number of video refresh frames, in this case, four. While one graphics calculation is in progress, the monitor is displaying the results of the last calculation. The ZOH represented by this 4 frame hold of the image gives a 2 frame delay, T3.

CONCLUSION

This measurement technique was used to verify the expected simulation transport delay. The method involved very simple software modifications. The display model proved to be of significant importance in accurately measuring the transport delay through a multi-refresh graphics image.

Improvements in this technique are ongoing. Multiple luminance level sensing has successfully reduced noise levels for single frame update graphics. Multi-frame graphics at many luminance levels requires a sample-and-hold that is synchronized with the screen refresh. This is a current project.

TABLE 1. Transport Delay Measurement Data

Frequency (Hz)	Phase (Degrees)	Phase Due to Dynamics (Degrees)	Measured Transport Delay (mSec.)	Expected Delay (mSec.)
0.25	-69.7 -71.2 -68.9 -69.5 avg. -69.8	-59.1	119	117
1.15	-162.1 -161.1 -160.8 -159.5 avg. -160.9	-111.2	120	117
1.87	136.2 136.9 137.6 140.8 avg. 137.9 = -222.1	-139.7	122	117
3.09	71.0 66.3 69.2 65.1 avg. 67.9 = -292.1	-157.1	121	117

REFERENCES

1. Riccio, G.E., Cress, J.D., & Johnson, W.V. (1987). The Effects of Simulator Delays on the Acquisition of Flight Control Skills: Control of Heading and Altitude. Proceedings of the 31st Meeting of the Human Factors Society. New York, NY:Human Factors Society.

2. Gum, D.R., & Martin, E.A. (1987, August). The Flight Simulator Time Delay Problem. AIAA paper 87-2369-CP, Monterey, CA

3. Kuo, Benjamin C. (1963). Analysis and Synthesis of Sampled-Data Control Systems. Englewood Cliffs, NJ : Prentice-Hall, Inc.

DETERMINATION OF HELICOPTER SIMULATOR TIME DELAY AND ITS EFFECTS ON AIR VEHICLE DEVELOPMENT*

John Woltkamp**
S. Ramachandran, PhD†
Roger Branson‡

McDonnell Douglas Helicopter Company
Mesa, Arizona 85205

Abstract

McDonnell Douglas Helicopter Company recently undertook a study to determine the actual simulator hardware time delay in all the simulators. It also investigated the effect of time delay on pilot performance and his aircraft evaluation in an engineering design environment. This paper describes the system architecture, techniques of measuring thruput delays, and initial study results. It was found that the average total system delay was about 87 milliseconds, well below values reported in open literature for most of the training and engineering simulators. The second phase of the study involved systematically varying the simulator delays so that data on the effect of time delay could be collected and used as a useful parameter in aircraft/simulator design. Pilot performance was recorded and subjective evaluations in the form of Cooper-Harper ratings were also obtained. Analysis of pilot performance did not provide any dramatic changes due to increased simulator delays but did show that the pilot control activity increased in the low speed, high gain tasks. It was found that with increased time delay the Cooper-Harper rating increased indicating degradation in perceived handling qualities. However, for the type of helicopter simulated, there was not a definite time delay at which the ratings changed abruptly. This indicates that for engineering design purposes while it is desirable to keep the delay to the absolute minimum, there may be sufficient flexibility in the design of the simulator to permit cost/capability trade offs. However, this needs to be further validated by additional tests that introduce pilot distractions (such as gusts) and force the pilot to increase his closed loop gain.

Abbreviations

A/D	Analog to digital
CIV	General Electric CompuScene IV
COMM	Communications
CPU	Central processor unit
DIST	Distribution
ETSD	Engineering and Training Simulation Department
HMT	Helmet mounted tracker
HSD	High Speed Data Interface
Hz	Hertz
IAT	Image auto-tracker
IHADDS	Integrated Helmet and Digital Display System
IPU	Internal processor unit
I/O	Input/output
MFD	Multi-functional display
mm/Sec	Millimeters per second
msec	Milliseconds
RTIO	Real-time input/output
SCS	Systems control station
SOPS	Servo Optical Projection System
VSM	Video/Switcher Mixer

Introduction

With the advances in simulation technology, simulators have come to play a very important role in the design and development of aircraft [References 1-4]. While the use of simulators for fixed wing design has been more common, engineering simulation as a design tool in rotorcraft development is a fairly recent application [References 5-16]. The simulation fidelity is very much dependent on a number of factors and there have been a number of research articles on this [References 17-23]. One if the important factors that influences simulation fidelity and pilot's perception as well as his ability to control and fly the air vehicle is simulator time delay. A number of researchers have examined this issue predominantly from a fixed wing aircraft aircrew training perspective [References 24-26]. The Naval Training Systems Center (NTSC) has conducted several studies on helicopter simulation fidelity [References 27-30]. These papers have examined in general terms the influence of visual system delay, among other factors, in hover/landing training tasks. The simulators used in these studies were, of necessity, limited in capability especially in visual cues and had somewhat large simulator system delays. Much more advanced helicopter simulators with superior visual simulation realism have been recently developed by several helicopter manufacturers. These simulators have come to play a very important role in the design of helicopters. These considerations make it necessary to examine in depth what role simulation time delay plays in an engineering design environment.

McDonnell Douglas Helicopter Company had recently set up a multi-ship simulation capability in support of its rotorcraft design and development [References 9, 14, 16, 31]. This facility currently has three twenty-foot diameter dome, fixed-base simulators. The facility employs state-of-the-art computational and visual systems with head-tracked, very large field-of-regard display systems. Modular crew stations simulating the appropriate air vehicle configurations are set up in the domes for engineering evaluations. As a part of simulation validation, the simulation department undertook efforts to determine the magnitude of simulator system delay. It was also desired to assess what impact the simulator system delay plays in a new aircraft design where the actual aircraft has not yet been flown and whose actual flight characteristics are unknown to the pilot. It may be noted that to date there has been no definitive study to

*For presentation at the AIAA Flight Simulation Technologies Conference, September 1988.
**Member Technical Staff
†Senior Project Manager, Member AIAA
‡Chief, Flight Dynamics and Controls Simulation

establish system delay requirements for helicopter simulators and the helicopter manufacturer has to use engineering judgment in the simulator application for design.

The first part of the paper discusses the determination of actual system delay in the McDonnell Douglas Helicopter Company simulators. It briefly describes the simulator system, the test set up and methodology and test results. The second part describes piloted simulator evaluation of a generic, high performance, combat helicopter with varying simulator time delays. It describes the test maneuvers, the test pilots' background and their assessment of aircraft handling characteristics.

System Delay Measurements

Scope

The simulator system delay is defined here as the time interval between a step input at the crew station flight controller and a resultant change in the visual sensory cue output. Two sources contribute to the total system delay. First is the finite time delay (thruput delay) involved in hardware implementation of simulation of the aircraft and its system/subsystems, and the aircraft's relation to its environment. The second source is model inexactness arising from approximating a complex physical process for ease of real-time simulation. If a perfect model is used this delay will be zero and the system delay will be just the hardware thruput. This study assumed a perfect model because of the generic nature of the simulated aircraft. Hence, the system time delay measured is strictly the hardware delay.

The study made the following additional assumptions:

1) Frames were not overrun [later verified].

2) Transmission times were small in magnitude with respect to the total thruput delay, i.e., for hardware interfaces, electrical signals and light propagation.

Description of the System

The system used for the study was a known, managed configuration of hardware and software modules. Fig. 1 illustrates the simulator hardware architecture. This system is hosted on a Gould Concept 32 Series 9780 computer system that runs the simulation in several internal tasks in two separate processors: 1) a Central Processor Unit (CPU), and 2) an Internal Processor Unit (IPU). This study concentrated on three specific tasks: 1) the control loader interface task (running in the CPU), 2) the IPU task, and 3) the CPU task. The control loader interface task accepts the control stick inputs from the HSD link and places the control positions into DATAPOOL common memory locations. The IPU has responsibility for running the flight model and control laws. The IPU cannot do I/O and shares memory with the CPU through DATAPOOL. DATAPOOL is a feature of the Gould operating system similar to a FORTRAN common with a non-relocatable memory partition assigned during system configuration. In the IPU task, the control positions are retrieved from DATAPOOL, scaled and offset, and then used by the aero model to compute and update the aircraft position and attitude, which are then stored in DATAPOOL memory locations. The CPU does all I/O tasks. Specifically, the CPU task is responsible for: 1) digital image generation system interface, 2) avionics interface, 3) systems control station (SCS) interface, 4) environment control, 5) initialization, 6) data collection, and 7) mission record/ playback. All tasks in the IPU and the CPU communicate through DATAPOOL common memory. The simulator is configured with a conventional flight controls system.

The simulator interfaces to a separate General Electric CompuScene IV (CIV) digital image generation system, hosted on a Gould Concept 32 Series 9780 computer system, referred to as a Frame I. The output of the visual system is routed through a video matrix switcher/mixer to the background and inset General Electric light valve projectors. The projector images are routed through a Servo Optical Projection System (SOPS) that optically combines the images, and positions the projection lens through servo control in either a fixed forward mode or in head-tracking mode.

The conventional flight controls system consisted of a McFadden hydraulically driven, control loading system with a digital interface to the real-time host. The force versus displacement feel characteristics model replicated the feel characteristics of a generic rotorcraft.

Fig. 2 illustrates the conventional flight controls system data flow for the three tasks with respect to cockpit control signals in and the FRAME I signal out. We have not included secondary flow paths used for monitoring, external control or environment control such as those related to the audio/communication link, the system control station/instructor inputs, and the sensor channel control.

The diagram does not show that the CPU, IPU, and CIV tasks are running asynchronously and that each task is complete inside of its frame respectively. The three main tasks ran at a 60 Hz frame rate on both systems. The CIV only displays attitude and position. The CIV does not extrapolate position or attitude based on rate.

The other routines that should be mentioned are real-time data save to tape and the strip recorder output. On digital tape, we collected data for raw stick (digital) input, shaped and filtered stick inputs, aero forces and moments, position, accelerations, CIV interface buffered data, and a limited number of performance variables and system control flags. This is done from the CPU at 30 Hz. The strip recorder data included the analog stick position and force. The data collected on both the strip chart recorder and save tape are asynchronous.

Thruput Test Cases

The total thruput delays are relative to step inputs made at the cockpit [or controller card] and to the response of the CIV monitor. The normal path of data flow for the conventional flight control systems is identified in Fig. 2: controller input, through the A/D, through the control loader interface task, through common memory, through stick scaling and offset, through the IPU task, through common memory, through the CIV interface task, through the Frame I and CIV computing systems, and output video for projection to the dome and repeat on a monitor.

For the purposes of testing the hardware thruput delay, the output from the IPU task of the normal data flow was ignored. Values of the flight control stick position placed into DATAPOOL by the interface task

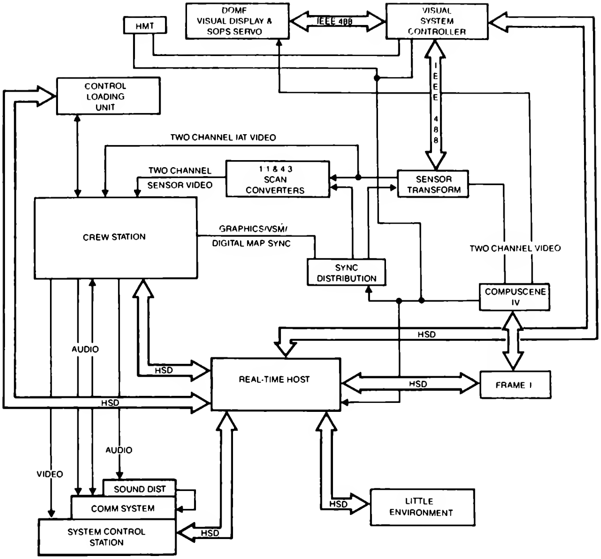


Fig. 1 System block diagram.

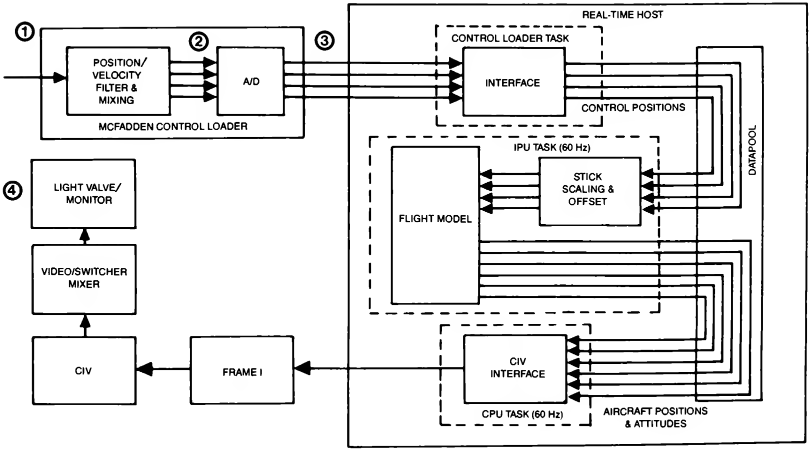


Fig. 2 System data flow.

were used as direct inputs to the CIV interface task. The CIV interface task monitored the flight control stick position variables for change and responded immediately in response to the stick position change. Modifications were made to the CIV interface task on each system for each test of a commanded axis, specifically pitch, roll, yaw, and lift.

Test Methodology for System Thruput Measurement

For each test case identified by input /response, a total thruput delay and a series of partial thruput delays (delay between simulator subsystems) were measured. Partial thruput delay was measured by trapping on changes to a preselected and identifiable memory locations via DATAPOOL common memory, line signal or generated interrupt. Digital data was collected from common memory and written to an on-line file and later dumped to tape for off-line processing. The digital data was collected at the CPU frame rate. Line signals were collected on strip charts running at 500 mm/Sec.

Total thruput delay was measured using defined pixel(s). A sensitive photo sensor was built to detect response of the visual system by sensing the change in pixel brightness. Analog output from the photo sensor device was assigned to a strip chart recorder channel along side the controller analog signal channels. The defined pixel(s) for the test cases were chosen to be located in objects of dark, high contrast value, and in specific areas of the database such that rotating 90 degrees in the plane of the tested axis locates another pixel of bright, high contrast value, specifically the change in brightness must be detectable by the photo sensor. Objects selected were dark pylons that contrasted well with the light blue sky.

For the total hardware thruput measurement, the CIV interface task was modified so that any detected change in the stick input resulted in an in-plane rotation of the visual by 90 degrees. The aero model would continue to integrate a response, but the output was ignored since the model delay was not being measured as part of the hardware delay. Thus, for example, during the yaw axis test, the aircraft was initialized in front of the pylon. The photo sensor was then placed on the display to detect the dark pylon. When the stick input was detected, the CIV was sent an immediate (within one frame of detection) 90 degree heading change, which now placed the bright sky in view of the photo sensor.

Response of the photo sensor was found to be less than approximately 0.25 msec, and was considered insignificant. Response was also measured across the controller and was found to be less than 1 msec, highest resolution of the strip chart recorder being used. We were unable to intercept electrical signals from within the control. We decided to monitor the output signal and monitored those signals at output test connections on the controller card. Another feature of the controller card allowed us to step the input voltage, similar to the result achievable to a square wave generator. We used the voltage step as our input for the conventional flight controls system once the aircraft was in position.

Total Simulator Response Test

These tests measured visual response time with respect to a step input. The flight model response is bypassed so that pure hardware delay can be measured. The tests were conducted as two sets. The purpose was to spike the visual as soon as a control input was detected. One set of tests was completed with the pilot activating the controls in the cockpit. The second set was activated by switching input voltage to the card representing the control input. This technique is similar to the one used by NTSC [Reference 32]. A modification was made to the real-time host software which responded as soon as the input was detected. The response rotated the model position 90 degrees. The position change sent to the Frame I resulted in a visual change from the CIV.

The aircraft was positioned in a reproducible location in front of a vertical bar at the runway in the visual database. Once the correct position was established, the test engineer placed the photocell on the high resolution monitor in the most sensitive position then directed the pilot or engineer to make the appropriate step input in the directional axis using the control for one set of tests and the voltage switching for the second set of tests. The input control position signal and the photo sensor output signal were recorded at a rate of 500 mm/Sec. The time delay between reference points 1 and 4 [Fig. 2] was measured on the strip chart recorder.

Results

Fig. 3 summarizes the end-to-end results and shows an approximate total delay of 87 msec. This corresponds to path from reference points 1 to 4 in Fig. 2.

Effects of Simulator Time Delay

Having determined the simulator time delay, the next phase of the study was to investigate how the time delay affects pilot's performance and his perception of the aircraft. The simulator with conventional control was chosen for this study since the test pilots were more familiar with similar aircraft. This also avoided the digital control laws smoothing or masking the finer effects due to the time delay.

Test Method

The study required the pilots to fly three different types of courses/maneuvers; narrow slalom/dolphin, serpentine, and longitudinal quickstop. These three courses/maneuvers cover most of the helicopter's flight envelope; high speed, low speed-hover, and transition.

The Narrow Slalom/Dolphin course, Fig. 4, required the pilot to fly through pylons while maintaining constant speed and an altitude below 25 feet except when maneuvering over the two 50 foot obstacles within the course. Prior to each run the aircraft was initialized in a hover some distance out away from the course, the pilot then accelerated to his test speed of 70-85 knots. He entered the course at this speed and negotiated the pylons and 50 foot obstacles. This task heavily taxed his lateral, directional, and vertical controls while attempting to maintain constant velocity. The Serpentine course, Fig. 5, was a low speed maneuvering task of 10-20 knots. The pilot was required to start from a stabilized hover then follow the centerline of the closed course for one lap. The pilot must follow the course, stay below 25 feet, and maneuver over two 25 foot brick wall obstacles. The run was complete after one lap. This course required precise control in all axes.

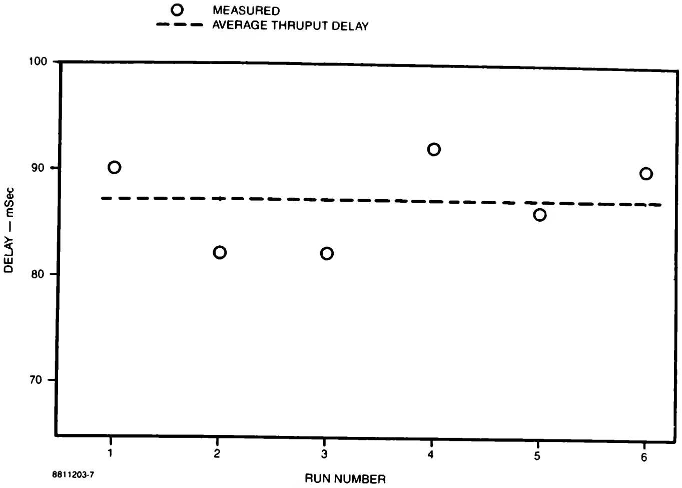


Fig. 3 Total thruput delay.

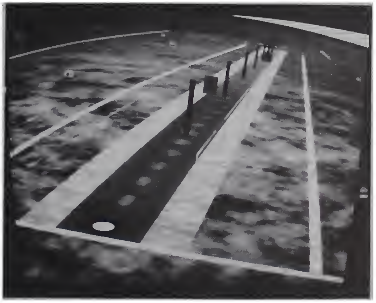


Fig. 4 Narrow slalom/dolphin course.

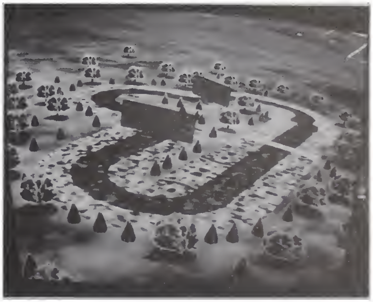


Fig. 5 Serpentine course.

·The longitudinal quickstops were done along a road next to the Slalom/Dolphin course. This is visible on either side of the course (Fig. 4). From a stabilized hover the pilot accelerated to 60 knots, then decelerated back to a hover while maintaining constant altitude. He was required to maintain his alignment over the road, constant heading, and altitude. The run was complete when a hover was again established.

The helicopter crewstation, as mentioned earlier, was conventionally configured with cyclic, pedals, and collective with an integrated helmet and digital display system (IHADDS) for heads up flight symbology. The flight symbology displayed ground speed, heading, radar altitude, torque, and rate of climb simultaneously. Below 20 knots, an acceleration cue and velocity vector were displayed, while above 20 knots, a pitch ladder appeared showing pitch and roll attitude.

Delay was added to the pilot inputs in the real-time host software at the point the stick scaling and offsets were calculated [see Fig. 2]. The input data for all axes from the McFadden control loader task was stored in separate arrays along with 2 seconds of past data which was constantly updated, similar to pushing data onto a stack. The software control was written such that the simulation operator could select 0 to 10 frames of delay to the control law inputs, which determined the point at which data was pulled from the stack. The delay was added equally to all axes of control: longitudinal cyclic, lateral cyclic, pedal position, and collective.

The pilot was exposed to 0, 2, 4, 6 and 10 frames of delay to input; at 60 Hz this is 16.7 msec/frame. Prior to taking data the pilot was allowed to practice the course until he was comfortable with the task. When ready he was initialized at the start point and proceeded to fly the run. He flew each delay 3 times in a somewhat random order for a total of 15 runs. The tests on each course took approximately 40 minutes to complete. During each run, performance data was recorded on magnetic tape and analyzed at the end of the session for controller activity. At the end of each run, the pilot was asked to evaluate the task with Cooper-Harper ratings [Fig. 6] [Reference 32]. Sessions were limited to one and one half hours per day per pilot, and only one course was flown each day to alleviate pilot fatigue.

Pilot Backgrounds

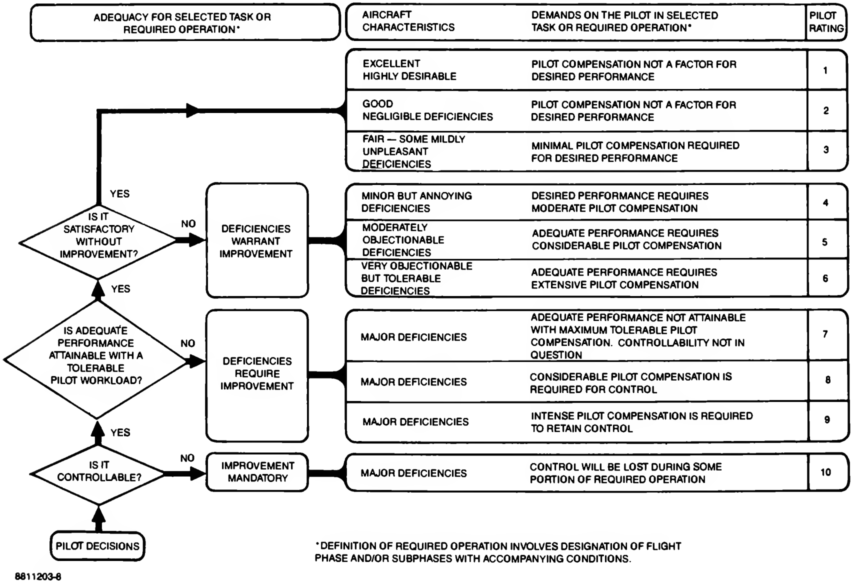
We selected four pilots as subjects for the delay study. Total flight time for each pilot ranged from a low of 2650 to a high of 7300 hours, Pilots A, C, and D were former U.S. Army pilots. Pilot B was a former U.S. Marine Corps pilot. All pilots are present employees of McDonnell Douglas Helicopter Company. Pilots A and B are company experimental test pilots. They had test pilot school training and were well versed in conduct of similar studies. Pilot C is a flight controls engineer and a current instructor pilot with the Army National Guard. Pilot D is a simulation pilot /engineer. Pilot A has 5300 hours of total rotary wing flight time, Pilot B 1950 hours, Pilot C 2950 hours, and Pilot D 3250 hours. Their backgrounds included nap-of-the-earth flight experience and scout /attack mission experience.

Each of the pilots had expressed concern for and the importance of duplicating the real vehicle delay and minimizing the simulation delay. They felt that this fidelity factor of simulation was critical to any development program or study of handling qualities and performance.

Each pilot was briefed about the control input delays during the study. The sequence of delays would be random, but would remain constant for each run of the course. They were asked to evaluate the task using the Cooper-Harper rating scheme, and not to try to guess the delay length.

Results

Fig. 7 is a graphic presentation of the ratings plotted against the delay for each course. From the median of the ratings, a slight trend moving from "deficiencies warranting improvement" to "deficiencies requiring improvement" is indicated with increased time delay. While the vehicle was still controllable, the pilots could not estimate the number of frames at which the simulation would become uncontrollable. The subjective Cooper-Harper ratings confirm their impressions.


ADEQUACY FOR SELECTED TASK OR REQUIRED OPERATION*
AIRCRAFT CHARACTERISTICS
DEMANDS ON THE PILOT IN SELECTED TASK OR REQUIRED OPERATION*
PILOT RATING
EXCELLENT HIGHLY DESIRABLE
PILOT COMPENSATION NOT A FACTOR FOR DESIRED PERFORMANCE
1
GOOD NEGLIGIBLE DEFICIENCIES
PILOT COMPENSATION NOT A FACTOR FOR DESIRED PERFORMANCE
2
FAIR — SOME MILDLY UNPLEASANT DEFICIENCIES
MINIMAL PILOT COMPENSATION REQUIRED FOR DESIRED PERFORMANCE
3
YES
IS IT SATISFACTORY WITHOUT IMPROVEMENT?
NO
DEFICIENCIES WARRANT IMPROVEMENT
MINOR BUT ANNOYING DEFICIENCIES
DESIRED PERFORMANCE REQUIRES MODERATE PILOT COMPENSATION
4
MODERATELY OBJECTIONABLE DEFICIENCIES
ADEQUATE PERFORMANCE REQUIRES CONSIDERABLE PILOT COMPENSATION
5
VERY OBJECTIONABLE BUT TOLERABLE DEFICIENCIES
ADEQUATE PERFORMANCE REQUIRES EXTENSIVE PILOT COMPENSATION
6
YES
IS ADEQUATE PERFORMANCE ATTAINABLE WITH A TOLERABLE PILOT WORKLOAD?
NO
DEFICIENCIES REQUIRE IMPROVEMENT
MAJOR DEFICIENCIES
ADEQUATE PERFORMANCE NOT ATTAINABLE WITH MAXIMUM TOLERABLE PILOT COMPENSATION. CONTROLLABILITY NOT IN QUESTION
7
MAJOR DEFICIENCIES
CONSIDERABLE PILOT COMPENSATION IS REQUIRED FOR CONTROL
8
MAJOR DEFICIENCIES
INTENSE PILOT COMPENSATION IS REQUIRED TO RETAIN CONTROL
9
YES
IS IT CONTROLLABLE?
NO
IMPROVEMENT MANDATORY
MAJOR DEFICIENCIES
CONTROL WILL BE LOST DURING SOME PORTION OF REQUIRED OPERATION
10
PILOT DECISIONS
8811203-6

*DEFINITION OF REQUIRED OPERATION INVOLVES DESIGNATION OF FLIGHT PHASE AND/OR SUBPHASES WITH ACCOMPANYING CONDITIONS.

Fig. 6 Cooper-Harper rating (from Reference 33).

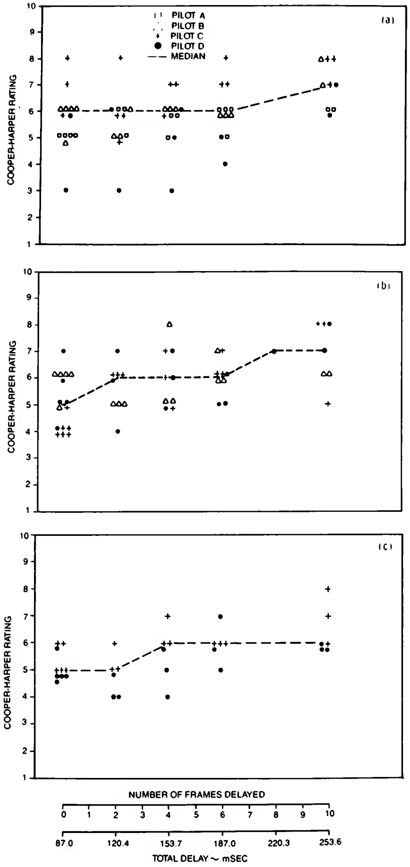


Fig. 7 Effect of time delay on Cooper-Harper ratings: a) narrow slalom course, b) serpentine course, and c) longitudinal quick stop.

Reviewing the plotted digital data, we found that the delay was more significant in the low speed, high gain task. During the high speed task, we found that as the delay increased, the stick activity increased only slightly, but the flight path became slightly tighter past the pylons. An example plot is shown in Fig. 8. The data also confirmed that the tasks were still controllable with only a slight increase in activity as the delays were increased from 0 to 10 frames.

Conclusion

McDonnell Douglas Helicopter Company had recently set up an advanced rotorcraft simulation facility. As a part of the simulation validation process, the simulator time delay was measured. The test consisted of applying pilot control input at the stick and measuring time delay to obtain the visual system response. Measurements showed that the average thruput delay was approximately 87 msec. This is significantly less than the time delays reported in the literature for other simulators. As a second part of the study, piloted evaluations were done to ascertain the impact of time delay on pilot's performance and their evaluations of the aircraft. A generic high performance helicopter was simulated and typical maneuvers covering important regions of the flight envelope were flown. Four pilots with high helicopter flight time served as study subjects. Varying time delays (up to 253 msec) were introduced in a random fashion during different runs. Pilot performance was recorded and subjective evaluations in the form of Cooper-Harper ratings were also obtained. Analysis of pilot performance did not provide any dramatic changes due to increased simulator delays but did show that the pilot control activity increased in the low speed, high gain tasks. It was found that with increased time delay the Cooper-Harper rating increased indicating degradation in perceived handling qualities. However, for the type of helicopter simulated, there was not a definite time delay at which the ratings changed abruptly. This indicates that for engineering design purposes, while it is desirable to keep the delay to the absolute minimum, there is sufficient flexibility in the design of the simulator to permit cost/capability trade offs.

However, this judgment has to be validated by further studies based on the findings of Smith and Bailey for fighter aircraft in-flight simulation [Reference 34]. Their study cites instances where, with high initial time delay, the pilot performance degraded significantly when high stress level was introduced. It will be important to find out whether such performance degradation will be true for helicopters also and whether such effects can be reproduced in ground based helicopter simulators.

Acknowledgments

The authors thank Tom Galloway, Steve Butrimas and Blair Browder of Naval Training Systems Center for their review of this paper. We benefited considerably from their experience and suggestions.

References

1. Pedestarres, M., "L'Apport de la Simulation Dans le Developpement des Airbus," AGARD Conference Proceedings No. 424, May 1987.

2. Copeland, J.L., "Research Through Simulation," NASA Langley Research Center NF-125, February 1982.

3. Lypaczewski, P.A., Jones, A.D., and Voorhees, J.W., "Simulation in Support of Advanced Cockpit Development," AIAA Flight Simulation Technologies Conference, AIAA Paper No. 87-2572, Monterey, CA, August 1987.

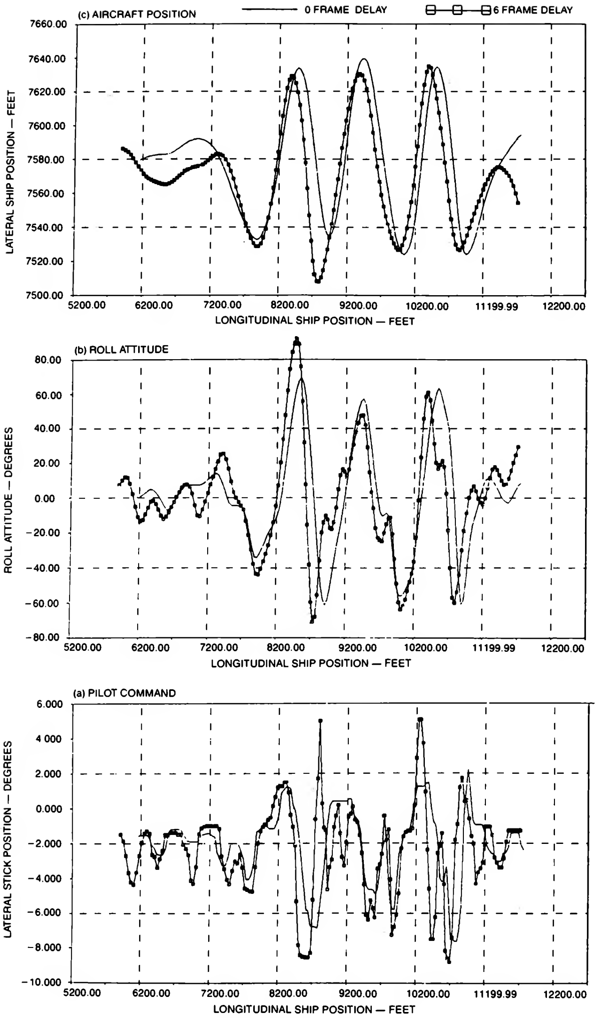


Fig. 8 Example of effect of time delay on commanded input and response.

4. Anon., "Research Cockpit Test Avionics," Jane's Defence Weekly, Vol. 9, No. 13, April 2, 1988, pg. 643.

5. Ramachandran, S., Richeson, W.E., and Borgman, D.C., "Reducing Rotary Wing Aircraft Development Time/Cost Through the Use of Simulation," AGARD Conference Proceedings No. 424, May 1987.

6. Buckingham, S.L., "Helicopter Flight Control Research – a Demanding Application of Piloted Simulation," Proc. Conference on Flight Simulation of Helicopters Status and Prospects, Royal Aeronautical Society, April 1985.

7. Offenbeck, H.G., Dahl, H.J., "Helicopter Simulation for Design," Proc. Conference on Flight Simulation of Helicopters Status and Prospects, Royal Aeronautical Society, April 1985.

8. Bulban, E.J., "Bell Sees the 21st Century Through Simulation," Rotor and Wing International, August 1987, Vol. 21, No. 9, pp. 46-58.

9. Anon., "Enhanced Simulators Aid Development of Engineering, Man/Machine Interfaces," Aviation Week and Space Technology, January 19, 1987, Vol. 126, No. 3, pp. 56-58.

10. Anon., "NASA/Army Upgrading Simulation Facilities at Ames," Aviation Week and Space Technology, January 19, 1987, Vol. 126, No. 3, pg. 57.

11. Anon., "Human Factors Research Key Element of LHX Design," Aviation Week and Space Technology, January 19, 1987, Vol. 126, No. 3, pp. 118-122.

12. Toler, D., "NASA Simulators Join LHX Crew-Size Quest," Rotor and Wing International, July 1987, Vol. 21, No. 8, pp. 42-43.

13. Marsh, G., "Closing the Visual Reality Gap," Defence Helicopter World, December 1986, Vol. 5, No. 6, pp. 4-8.

14. Wilson, S., "Responding not Reacting," Defence Helicopter World, April 1988, Vol. 7, No 2, pp. 6-9.

15. Marsh, G., "Modelling Rotor Blades – the Way Forward?" Defence Helicopter World, April 1988, Vol. 7, No. 2, pp. 42-43.

16. Barber, J.J., "MDHC's Dual-Approach Philosophy to Simulation," Rotor and Wing International, March 1988, Vol. 22, No. 3, pp. 58-60.

17. Anon., Collection of papers, Advances in Flight Simulation – Visual and Motion Systems, International Conference Proceedings, Royal Aeronautical Society, London, May 1986.

18. Parrish, R.V., Houck, J.A., Martin, D.J., "Empirical Comparison of a Fixed-Base and a Moving-Base Simulation of a Helicopter Engaged in Visually Conducted Slalom Runs," NASA TN D-8424, May 1977.

19. Ferguson, S.W., et al., "Assessment of Simulation Fidelity Using Measurements of Piloting Technique in Flight," Paper presented at the AHS Forum, 1984.

20. Ferguson, S.W., et al., "Assessment of Simulation Fidelity Using Measurements of Piloting Technique in Flight – Part II," 41st Annual Forum of the AHS, Ft. Worth, TX, May 1985.

21. Morton, J.M., "Helicopter Data Requirements for Handling Fidelity of Training Flight Simulators," Proc. Conference on Helicopter Simulation sponsored by FAA/NASA, Atlanta, GA, April 1984.

22. Bray, R.S., "Experience with Visual and Motion Cueing in Helicopter Simulation," Proc. Conference on Helicopter Simulation sponsored by FAA/NASA, Atlanta, GA, April 1984.

23. Crampin, T., "Transport Delays, Texture and Scene Content as Applied to Helicopter Visual Simulation," Proc. Conference on Helicopter Simulation sponsored by FAA/NASA, Atlanta, GA, April 1984.

24. Crane, D.F., "Compensation for Time Delays in Flight Simulators Visual Display Systems," Proc. Conference on Helicopter Simulation sponsored by FAA/NASA, Atlanta, GA, April 1984.

25. Bailey, R.E., Knotts, L.H., Horowitz, Capt. S.J., and Malone, Capt. H.L. III, "Effect of Time Delay on Manual Flight Control and Flying Qualities During In-Flight and Ground-Based Simulation," AIAA Flight Simulation Technologies Conference, AIAA Paper No. 87-2370, Monterey, CA, August 1987.

26. Merriken, M.S., Riccio, G.E., and Johnson, W.V., "Temporal Fidelity in Aircraft Simulator Visual Systems," AIAA Flight Simulation Technologies Conference, AIAA Paper No. 87-2370, Monterey, CA, August 1987.

27. Ricard, G.L., Parrish, R.V., Ashworth, B.R., and Wells, Lt. M.D., "The Effects of Various Fidelity Factors on Simulated Helicopter Hover," NAVTRAEQUIPCEN IH-321, January 1981.

28. Galloway, R.T., "Training Research on Improved Navy Helicopter Simulators," Proc. Conference on Helicopter Simulation sponsored by FAA/NASA, Atlanta, GA, April 1984.

29. Westra, D.P. and Lintern, G., "Simulator Design Features for Helicopter Landing on Small Ships. I: A Performance Study," NAVTRASYSCEN 81-C-0105-13, April 1985.

30. Westra, D.P., Sheppard, D.J., Jones, S.A., and Hettinger, L.J., "Simulator Design Features for Helicopter Shipboard Landings. II: Performance Experiments," NAVTRASYSCEN 85-D-0026-9, To be published.

31. Harvey, D., "Building the Ideal Plant: MDHC's Spot in the Sun," Rotor and Wing International, Vol. 21, No. 1, January 9, 1987, pp. 46-47.

32. Butrimas, S. and Browder, B., "A Unique Approach to Specification and Testing of Simulators," AIAA Flight Simulation Technologies Conference, AIAA Paper No. 88-2570, Monterey, CA, August 1987.

33. Cooper, G.E. and Harper, R.P. Jr., "The Use of Pilot Rating in the Evaluation of Aircraft Handling Qualities," NASA TN D-5153, April 1969.

34. Smith, R.E. and Bailey, R.E., "Effect of Control System Delays on Fighter Flying Qualities," Agard Conference Proceedings No. 333.

HUMAN PERFORMANCE DATA IN SIMULATION DESIGN

88-4621-CP

Kenneth R. Boff, PhD
Human Engineering Division
Armstrong Aerospace Medical Research Laboratory
Wright-Patterson Air Force Base, OH 45433

Edward A. Martin, PhD
Training Systems Division
Aeronautical Systems Division
Wright-Patterson Air Force Base, OH 45433

Abstract

Reliable data on the aircrew's ability to acquire and process task critical information are of prime importance to the design of effective aircrew operational workstations and training devices and systems. While the available body of psychophysical research contains a staggering volume of human perceptual and performance data and principles that are of potential value to the designers, these are typically not systematically considered by the design engineer.[2,3] Though the nature and availability of these data are a key reason for this lack of systematic application, the problem can also be attributed to the basic skills and inclinations of designers, limitations in the available support environment, and to constraints imposed by the system design and acquisition processes.[6,18]

Introduction

There is a widely growing concern regarding the need to improve overall weapon system effectiveness by employing a more systematic consideration of the human element throughout the weapon system acquisition process. It is no longer considered sufficient to design to optimize equipment-related variables -- the system design process must consciously include human-related variables in design tradeoffs as well. Past emphasis on technology, resources, and schedule has led to problems which might have been avoided with a more complete consideration given the human's capabilities and limitations in the design process.[1] DoD initiatives are now requiring that human capabilities, limitations, and the training needed to achieve effective system operation be balanced against hardware design alternatives.[10,11,20] Weapon system effectiveness relies heavily on the capability of the crew to effectively interact with the equipment. This, in turn, depends upon the match between the bandwidth of the controls and displays and the sensory, perceptual, cognitive, and response characteristics of the human operator. Effective real-time simulation is an indispensable design, evaluation, and training tool to aid in optimizing this match.

Simulation is widely used to support the research and empirical evaluations of the human interface which assist design decisions during development, as well as the training necessary to fine-tune the crewmembers' information processing and control capabilities to that of the system. However, flight simulators cannot be fully effective in either the design or training environment if task-critical information germane to the research or training objectives is missing or inappropriately displayed;[14] indeed, improper or incomplete display of task-critical information in the simulation may have disastrous consequences if the behavioral effects of that information mismatch are not adequately accounted for in the specific application.[19] It is essential that the simulator designer be cognizant of the human's ability to process information so that the displays can be tailored to the simulation objectives, and that the behavioral implications of any spurious or missing task-critical information can be identified.[2] There exists a broad base of knowledge regarding human capabilities and limitations which can be of value to the designer in this regard. In fact, the volume of human performance and perceptual data of potential value to the simulation designer is overwhelming. Unfortunately, it is also widely fragmented and often difficult for the designer to interpret.[6] This situation is further exacerbated by the low perceived value of these data by the design community -- a perception fueled by both negative experiences with these data and the pressure to account for a myriad of other system-related variables in the course of the design process.[13,16] Thus, even as the need for reliable data regarding the human's capability to acquire, process, and respond to information is recognized, a variety of cross-disciplinary choke-points prevents designers from identifying, accessing, and applying the relevant information. The more significant barriers are the volume of information, its fragmentation across disciplines, and the conventions by which research results are packaged in the scientific community.[6]

Recognizing the need to facilitate the designer's accessibility, interpretability, and applicability of human performance and perceptual data, the Armstrong Aerospace Medical Research Laboratory established the Integrated Perceptual Information for Designers (IPID) project in 1980. This project has approached the problem of developing a reliable and utilitarian designers' resource of human-related data from three perspectives: (1) identification and consolidation of existing data of potential usefulness and a method to facilitate the transfer of these data to engineers, (2) identification of the system designer and the designer's biases regarding the use of human performance data, and (3) identification of the media best suited to communicate the information most effectively in the design environment.

The Integrated Perceptual Information For Designers (IPID) Project

IPID is a multi-agency effort supported by organizations within the U.S. Air Force, Army, Navy and NASA. Its prime objective is to provide "high-value" human performance data as a "low-cost" resource (in terms of risk, time and expense) to designers of operational crew systems and training devices. The project is organized around several interrelated information management and product objectives. Collectively, these objectives -- discussed in detail below -- are aimed at raising the perceived value and lowering the perceived cost of using human performance data in system design.[3,7]

Consolidating Existing Data

The identification and consolidation of human performance data of potential value to system design has involved the detailed treatment by sixty recognized experts of forty-five research disciplines. The resultant professional level reference work, the Handbook of Perception and Human Performance is a two-volume work of approximately 2800 pages.[4,5] It is illustrated with over 1600 figures and makes extraordinary use of data functions and schematics to present technical material. Data are plotted in standard units based on the Systeme Internationale. While the intended user of the Handbook is the ergonomist or research scientist or engineer, it provides a reliable basis for subsequent products of more direct design relevance. (See Table 1).

Packaging the Data to Facilitate Use

The presentation of these data was "human factored" to enable their direct use by system designers. The four volume Engineering Data Compendium has been designed as a primary reference for system designers of human-system interfaces.[9] It provides comprehensive information on the capabilities and limitations of the human operator, with special emphasis on those variables which affect the operator's ability to acquire, process and make use of task-critical information. Information was selected for inclusion into the Compendium on the basis of its practical potential for system design through an iterative process of review and analysis employing hundreds of technical subject matter experts and designers. Prospective entries were reviewed on the basis of statistical and methodological reliability, applicability to the normal adult population, and potential relevance to design problems.[13] The Compendium consists of approximately 1150 concise two-page entries designed to be self-contained, with information from related studies summarized and presented in graphic form, wherever possible. (See Figure 1 and Table 2). This reference will also be made available in Hypermedia format on a compact disk (CD-ROM) in 1988.

Training and Sensitizing Designers

Educational opportunities for training and sensitizing system designers to the value and application of human performance data in the design of operational and simulated controls and displays were developed and sponsored. The first of a series of short courses and seminars on "Engineering For Man-Machine Systems: Human Performance For System Designers" was conducted under the auspices of The University of Dayton in Dayton, Ohio, during June 1986. Offered again in June 1988 in Dayton, Lisbon (Portugal), Athens (Greece), and Delft (The Netherlands), its primary goal has been to provide system designers with a human performance framework for "decomposing" equipment-related design problems while sensitizing participants to the issues and approaches in using human performance data in human-machine system design.

Updating Information and Supporting Designers

An institution with responsibility for maintenance, update, and real-time support for analysis of the existing data with respect to system design problems will be established in 1988. In conjunction with the Tri-services, NASA, and NATO AGARD, the Armstrong Aerospace Medical Research Laboratory will establish and host a Crew System Ergonomics Information Analysis Center (CSERIAC). CSERIAC will provide a full range of technical information services in support of crew systems research, design and development in the government, industrial, and academic sectors.[12] The essential mission of CSERIAC is to maintain contact with the relevant knowledge and experience bases across these sectors and to develop the ability and media to draw upon and focus this expertise to solve problems, achieve expert consensus and to aid planning for more effective use of ergonomics data in the system design process. In addition, CSERIAC will provide information services, including topical reviews, special analysis reports, data, models, design support software, methodological assistance and a "Current Awareness" newsletter. Maintenance and update of data bases, including the IPID Engineering Data Compendium will also be a function of CSERIAC. (See Table 3).

Information Access in the Design Environment

Exploratory research is being conducted to define and evaluate requirements for an automated design support capability which will aid system designers to efficiently access and tradeoff design-relevant technical information. Follow-on development of this "Designer's Associate" support system will eventually automate access of human performance data in context with other design-related information, while aiding designers to factor this information efficiently into design decision making.[8,15,17]

Summary

The modern flight simulator has evolved considerably as a tool to support both operational system design and training requirements. The design of operational and training systems poses demands on our knowledge regarding the limitations and capabilities of the human and how these relate to the human interface with the system. As system needs press the limits of total human performance capabilities, it becomes ever more imperative that designers adequately assess and predict workload and training demands imposed by system and mission requirements. Simulation can usefully serve designers in this regard -- however, the implications of the imperfect nature of the flight simulator as a surrogate aircraft must be recognized, whether it is employed as an engineering research device or as a component of a training system. Errors of omission, inclusion, and synchronization in the simulated environment can confound both empirical assessments of design alternatives and training effectiveness.[2,14,19] The extent to which designers (both specifiers and implementers) understand and account for these display errors in the simulation will significantly influence the effectiveness of the resultant weapon system. This requires sensitivity on the part of the designer to the variables which affect the operator's ability to acquire and respond to task-critical information.

The Engineering Data Compendium has been designed to be a reliable and comprehensive source of information on the capabilities and limitations of the human operator.[9] The Compendium is fully cross-referenced to the Handbook of Perception and Human Performance for the designer requiring more depth in a given topic area.[4,5] Together, these documents provide the most extensive available body of research findings on human perception, information processing, and perceptual-motor performance in a format useful to the engineering and simulation communities.

The availability of human performance data, through the Handbook and Compendium, is a necessary prerequisite to solving the general problem concerning the use of human performance data in simulation and design. However, mere availability of these data is not a sufficient condition to ensure their use. Operational and training system designers must be sensitized and trained in the use and value of these data in tradeoffs with equipment features. To support this need, IPID has set up a continuing series of workshops and short courses in conjunction with the University of Dayton. In addition, the CSERIAC will provide a mechanism for updating and supporting the application of these data to system design problems. Finally, Designer's Associate -- the most ambitious of the IPID programs -- is developing technology to facilitate a designer's access to human performance and training effectiveness data within the context of an automated design support environment which integrates these data with other design-relevant information and guidance. This design decision support system will significantly lower the cost and raise the efficiency of accessing, interpreting, applying, and evaluating the impact of these data in system designs.

References

1. A'Harrah, R.C. (1981). Results of a Survey on the Flying Qualities of Piloted Airplanes, Military Specification MIL-F-8785. NAVAIRSYSCOM.

2. Boff, K.R. and Martin, E.A. (1980). Aircrew information requirements in simulator display design: The integrated Cuing requirements study. Proceedings of the Second Interservice/Industry Training Equipment Conference, 355-362.

3. Boff, K.R., Calhoun, G. L., and Lincoln, J. (1984). Making Perceptual and Human Performance Data an Effective Resource For Designers. Proceedings of the NATO DRG Workshop (Panel 4). Royal College of Science, Shrivenham UK.

4. Boff, K.R., Kaufman, L. and Thomas J., Eds. (1986). Handbook of Perception and Human Performance: V.I: Sensory Processes and Perception. John Wiley and Sons, New York.

5. Boff, K.R., Kaufman, L., and Thomas, J., Eds. (1986). Handbook of Perception and Human Performance. V.II: Cognitive Processes and Performance. John Wiley and Sons: New York.

6. Boff, K.R. (1987). The Tower of Babel Revisited: On Cross-disciplinary Chokepoints In System Design. In "System Design: Behavioral Perspectives on Designers, Tools and Organizations". North-Holland: New York.

7. Boff, K.R. (1987). Designing for Design Effectiveness of Complex Avionics Systems: The Design, Development and Testing of Complex Avionics Systems. Las Vegas, NV: NATO Advisory Group For Aerospace Research and Development.

8. Boff, K.R. (1987). Matching Crew System Specifications to Human Performance Capabilities. Proceedings of the 45th NATO AGARD Guidance and Control Panel Symposium. Stuttgart, W. Germany.

9. Boff, K.R. and Lincoln, J. (1988). Engineering Data Compendium: Human Perception and Performance. (4 Volumes). Armstrong Aerospace Medical Research Laboratory, Wright-Patterson AFB, OH and NATO AGARD.

10. Gentner, F.C. (1987). What's Happening at ASD Regarding MPT. Proceedings of the Ninth Interservice/Industry Training Systems Conference, 342-350.

11. Hardman Development Office (1985). Hardman Methodology: Aviation. Hardman Pub. 84-02. Chief of Naval Operations: Washington DC.

12. Hennessy, R.T. and McCauley, M.E. (1987). Proposal and Justification to Establish a Department of Defense Crew System Ergonomics Information Analysis Center. Armstrong Aerospace Medical Research Laboratory Technical Report, Wright-Patterson AFB, OH.

13. Klein, G.A. and Brezovic, C. (1987). What do Training Device Designers want to know about Human Performance Characteristics? Armstrong Aerospace Medical Research Laboratory, Wright-Patterson AFB, OH. AAMRL-TR-87-010

14. Martin, E.A., McMillan, G.R., Riccio, G.E. (1986). A Program to Investigate Requirements for Effective Flight Simulator Displays. Royal Aeronautical Society Conference on "Advances in Flight Simulation Visual and Motion Systems", London UK.

15. Martin, E.A. (1987). Toward a more systematic, efficient design process. The potential impact of intelligent design aids. In "System Design: Behavioral Perspectives on Designers, Tools and Organizations". North-Holland: New York.

16. Meister, D. and Farr, D. (1966) The Utilization of Human Factors Information by Designers. Office of Naval Research, NONR Contract #4974-00.

17. Reitman,W., Weischedel, R., Boff,K.R., Jones, M. and Martino, J. (1985). Automated Information Management Technology: A Technology Investment Strategy. Wright-Patterson AFB, OH. Air Force Aerospace Medical Research Laboratory, Report No. AFAMRL-TR-85-042.

18. Rouse, W.B. and Boff, K.R. (1987). System Design: Behavioral Perspectives on Designers, Tools and Organizations. North-Holland: New York.

19. Smith, R.E. and Bailey, R.E. (1982). Effect of Control System Delays on Fighter Qualities. AGARD Flight Mechanics Panel Symposium on "Criteria for Handling Qualities of Military Aircraft", Fort Worth TX.

20. Soldier Material Systems (1987). Manpower and Personnel Integration (MANPRINT) in the Materiel Acquisition Process. Army Regulation 602-2. Headquarters, Dept. of the Army: Washington DC.

Table 1. Contents of the HANDBOOK OF PERCEPTION AND HUMAN PERFORMANCE

Volume I: Sensory Processes and Perception

Section I: Theory and Methods...*Lloyd Kaufman, Section Editor.* Overview. **1.** Psychophysical Measurement and Theory, *J. C. Falmagne.* **2.** Strategy and Optimization in Human Information Processing, *George Sperling and Barbara Ann Dosher.* **3.** Computer Graphics, *Herbert Freeman.*

Section II: Basic Sensory Processes I...*Donald I. A. MacLeod and James P. Thomas, Section Editors.* Overview, *James P. Thomas.* **4.** The Eye as an Optical Instrument, *Gerald Westheimer.* **5.** Sensitivity to Light, *Donald C. Hood and Marcia A. Finkelstein.* **6.** Temporal Sensitivity, *Andrew B. Watson.* **7.** Seeing Spatial Patterns, *Lynn A. Olzak and James P. Thomas.* **8.** Colorimetry and Color Discrimination, *Joel Pokorny and Vivianne C. Smith.* **9.** Color Appearance, *Gunter Wyszecki.* **10.** Eye Movements, *Peter E. Hallet.*

Section III: Basic Sensory Processes II...*Carl E. Sherrick and Roger W. Cholewiak, Section Editors.* Overview, Carl E. Sherrick and Roger W. Cholewiak. **11.** The Vestibular System, *Ian P. Howard.* **12.** Cutaneous Sensitivity, *Carl E. Sherrick and Roger W. Cholewiak.* **13.** Kinesthesia, *Francis J. Clark and Kenneth W. Horch.* **14.** Audition I: Stimulus, Physiology, Thresholds, *Bertram Scharf and Soren Buus.* **15.** Audition II: Loudness, Pitch, Localization, Aural Distortion, Pathology, *Bertram Scharf and Adrianus J. M. Houtsma.*

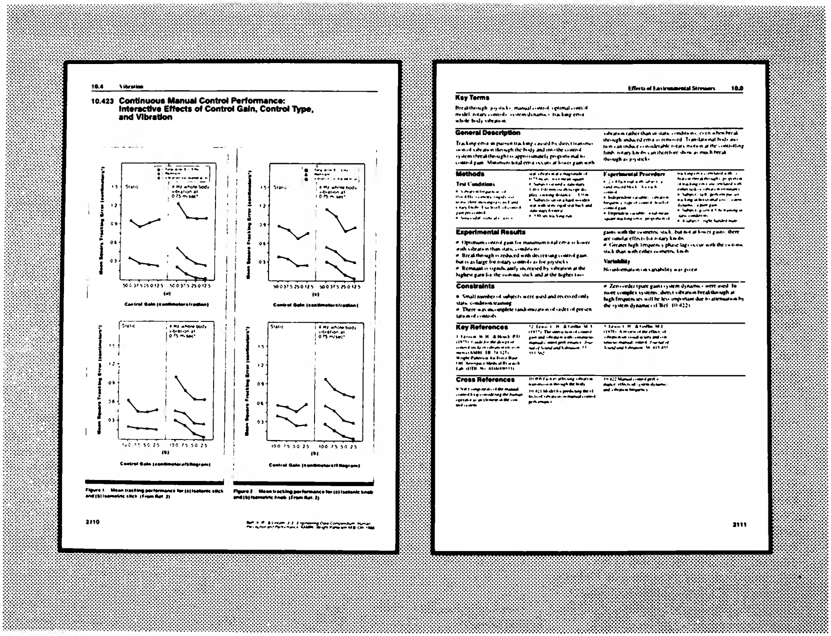
Section IV: Space and Motion Perception...*H. A. Sedgwick, Section Editor.* Overview, *H. A. Sedgwick.* **16.** Motion Perception in the Frontal Plane: Sensory Aspects, *Stuart Anstis.* **17.** Perceptual Aspects of Motion in the Frontal Plane, *Arien Mack.* **18.** The Perception of Posture, Self Motion, and the Visual Vertical, *Ian P. Howard.* **19.** Motion and Depth in Visual Acceleration, *David Martin Regan, Lloyd Kaufman, and Janet Lincoln.* **20.** Visual Localization and Eye Movements, *Leonard Matin.* **21.** Space Perception, *H. A. Sedgwick.* **22.** Representation of Motion and Space in Video and Cinematic Displays, *Julain Hochberg.* **23.** Binocular Vision, *Aries Arditi.* **24.** Adaptations of Space Perception, *Robert B. Welch.* **25.** Intersensory Interactions, *Robert B. Welch and David H. Warren.*

Volume II: Cognitive Processes and Performance

Section V: Information Processing...*Michael I. Posner, Section Editor.* Overview, *Michael I. Posner.* **26.** Auditory Information Processing, *Harold L. Hawkins and Joelle C. Presson.* **27.** Speech Perception, *Peter W. Jusczyk.* **28.** Visual Information Processing, *William G. Chase.* **29.** Perceiving Visual Language, *Thomas H. Carr.* **30.** Motor Control, *Steven W. Keele.*

Section VI: Perceptual Organization and Cognition...*Michael Kubovy, Section Editor.* Overview, *Michael Kubovy.* **31.** Tactual Perception, *Jack M. Loomis and Susan J. Lederman.* **32.** Auditory Pattern Recognition, *Diana Deutsch.* **33.** The Description and Analysis of Object and Event Perception, *I. Rock.* **34.** Spatial Filtering and Visual Form Perception, *Arthur P. Ginsburg.* **35.** Properties, Parts, and Objects, *Anne Treisman.* **36.** Theoretical Approaches to Perceptual Organization, *James R. Pomerantz and Michael Kubovy.* **37.** Visual Functions and Mental Imagery, *Ronald A. Finke and Roger N. Shepard.* **38.** Computational Aproaches to Vision, *H. G. Barrow and J. M. Tenenbaum.*

Section VII: Human Performance...*Jackson Beatty, Section Editor.* Overview, *Jackson Beatty.* **39.** The Effects of Control Dynamics on Performance, *Christopher D. Wickens.* **40.** Monitoring Behavior and Supervisory Control, *Neville Moray.* **41.** Workload: An Examination of the Concept, *Daniel Gopher and Emanuel Donchin.* **42.** Workload Assessment Methodology, *Robert D. O'Donnell and F. Thomas Eggemeier.* **43.** Vigilance, Monitoring, and Search, *Raja Parasuraman.* **44.** Changes in Operator Efficiency as a Function of Environmental Stress, Fatigue, and Circadian Rhythms, *G. Robert Hockey.* **45.** The Model Human Processor: An Engineering Model of Human Performance, *Stuart K. Card, Thomas P. Moran, and Allen Newell.* **Author Index. Cumulative Subject Index.**



10.4 Vibration

10.423 Continuous Manual Control Performance: Interactive Effects of Control Gain, Control Type, and Vibration

Static
4 Hz whole body vibration at 0.75 m/sec²
1.5
1.2
0.9
0.6
0.3
Mean Square Tracking Error (centimeters²)
50.0 37.5 25.0 12.5 50.0 37.5 25.0 12.5
(a)
Control Gain (centimeters/radian)
100 75 50 25 100 75 50 25
(b)
Control Gain (centimeters/kilogram)

Figure 1. Mean tracking performance for (a) isotonic stick and (b) isometric stick. (From Ref. 2)

Figure 2. Mean tracking performance for (a) isotonic knob and (b) isometric knob. (From Ref. 2)

2210

Ref: K. R. Boff & J. E. Lincoln. Engineering Data Compendium: Human Perception and Performance. AAMRL, Wright-Patterson AFB, OH, 1988.

Effects of Environmental Stressors 10.0

Key Terms

Breakthrough; joysticks; manual control; optimal control model; rotary controls; system dynamics; tracking error; whole-body vibration

General Description

Tracking error in pursuit tracking caused by direct transmission of vibration through the body and into the control system (breakthrough) is approximately proportional to control gain. Minimum total error occurs at lower gain with vibration rather than in static conditions, even when breakthrough-induced error is removed. Translational body motion can induce considerable rotary motion at the controlling limb; rotary knobs can therefore show as much breakthrough as joysticks.

Methods

Test Conditions

[illegible]

Experimental Procedure

[illegible]

Experimental Results

- Optimum control gain for minimum total error is lower with vibration than static conditions.
- Breakthrough is reduced with decreasing control gain, but is as large for rotary controls as for joysticks.
- Remnant is significantly increased by vibration at the highest gain for the isotonic stick and at the higher two gains with the isometric stick, but not at lower gains; there are similar effects for rotary knobs.
- Greater high-frequency phase lags occur with the isotonic stick than with either isometric knob.

Variability

No information on variability was given.

Constraints

- Small number of subjects were used and received only static condition training.
- There was incomplete randomization of order of presentation of controls.
- Zero-order (pure gain) system dynamics were used. In more complex systems, direct vibration breakthrough at high frequencies will be less important due to attenuation by the system dynamics (Ref. 10.422).

Key References

1. Levison, W. H., & Houck, P. D. (1975). Guide for the design of control sticks in vibration environments (AMRL-TR-74-127). Wright-Patterson Air Force Base, OH: Aerospace Medical Research Lab. (DTIC No. ADA008311)

*2. Lewis, C. H., & Griffin, M. J. (1977). The interaction of control gain and vibration with continuous manual control performance. Journal of Sound and Vibration, 55, 553-562.

3. Lewis, C. H., & Griffin, M. J. (1978). A review of the effects of vibration on visual acuity and continuous manual control. Journal of Sound and Vibration, 56, 415-457.

Cross References

9.503 Components of the manual control loop: considering the human operator as an element in the control system;

10.418 Factors affecting vibration transmission through the body;

10.421 Model for predicting the effects of vibration on manual control performance;

10.422 Manual control performance: effects of system dynamics and vibration frequency

2211

Figure 1. Sample Engineering Data Compendium entry.

Table 2. The ENGINEERING DATA COMPENDIUM as a design resource.

What is the Engineering Data Compendium?

The Engineering Data Compendium is a reference document which consolidates human sensory/perceptual and performance data in a form useful to system designers. It provides comprehensive information on the capabilities and limitations of the human operator, with special emphasis on those variables which affect the operator's ability to acquire, process, and make use of task critical information.

Who is the intended user?

The Compendium has been designed specifically for system designers who need an easily accessible and reliable source of human performance data. Scientists and engineers involved in research and development will also find the Compendium useful.

What assumptions are made concerning the user's expertise and background?

In the design of the Engineering Data Compendium, it was assumed that the user:

- will be reasonably sophisticated in the use of technical and quantitative data
- might have little prior training and experience in the specific technical areas treated in the Compendium
- recognizes the need to consider tradeoffs between human performance and equipment/environmental characteristics
- will be motivated to apply human performance data in system design

What types of information does the Compendium contain?

The Compendium consists of concise two-page data entries of the following types:

- basic human performance data
- section introductions outlining the scope of a group of entries and defining special terms
- summary tables integrating data from related studies
- descriptions of human perceptual phenomena
- models and quantitative laws
- principles and nonquantitative laws (nonprecise formulations expressing important characteristics of perception and performance)
- tutorials on specific topics to help the user understand and evaluate the material in the Compendium

What criteria were used in selecting information for the Engineering Data Compendium?

Data included in the Compendium were selected on the basis of:

- statistical and methodological reliability
- potential relevance to design problems
- applicability to the normal adult population

Limited material on human physiology and anatomy is included. Data from animal experimentation are omitted.

How is information organized and formatted in the Compendium?

- Information is presented graphically whenever possible, in the form of figures or tables.
- Text is organized into specific subsections such as General Description, Experimental Results, and Constraints. This format makes it easier for a Compendium user to locate or ignore specific items of information, by choice, and to assess the relevance of a given entry to a particular problem.
- Information in each entry addresses a relatively narrow topic. The goal is to provide information in discrete units that are easily understood by a user with little expertise in the topic area.
- Entries are designed to be as self-contained as possible.
- Information from related studies is summarized and integrated within individual entries wherever possible.

How will users locate the data they need?

Users will be able to select among structured approaches for accessing information depending upon how well the design issue has been defined. Besides a detailed table of contents, the Compendium features some innovative avenues by which the user can find the appropriate entry: key word indexes and glossaries on removable inserts, checklists addressing relevant human performance questions keyed to specific equipment design topics, and knowledge maps providing a relational hierarchy of the topic areas treated in the Compendium.

How will the Engineering Data Compendium be packaged?

The Compendium will consist of four volumes — three loose-leaf volumes of perception and performance data and a bound User's Guide containing supplementary aids, such as instructions for locating and using individual entries, design checklists, indexes, and a glossary. Multiple User's Guides are available for users who share a single set of data volumes.

Table 3. The CREW SYSTEM ERGONOMICS INFORMATION ANALYSIS CENTER (CSERIAC) as a design resource.

What are Information Analysis Centers?

The function of Information Analysis Centers (IACs) is to acquire, analyze, and disseminate specialized technical information according to expressed or anticipated needs. Department of Defense IACs are operated under directive of the Under Secretary of Defense for Research and Engineering. Most IACs are administratively managed and funded by the Defense Logistics Agency (DLA) and the Defense Technical Information Center (DTIC) and provide users with specialized engineering, technical, and scientific analytical services and products. Each IAC, staffed by scientists and engineers who are experts in their fields, focuses on a well-defined scientific or technical area. IACs are distinguished from technical information centers and libraries whose functions are concerned with providing reference or access to the documents themselves rather than the information contained in the documents. IAC subject coverage has greater breadth and depth than is possible from DTIC.

What is crew system ergonomics information?

Crew System Ergonomics information is scientific and technical knowledge and data concerning human characteristics, abilities, limitations, physiological needs, performance, body dimensions, biomechanical dynamics, strength, and tolerances. It also includes engineering and design data about equipment intended to be used, operated, or controlled by members of military crews.

What is the objective of CSERIAC?

The objective of CSERIAC is to efficiently support the requirements of the Department of Defense for incorporating crew system ergonomics in the design and operation of military systems. To achieve this objective, CSERIAC will establish a network among relevant knowledge sources on an international scale and develop the media to draw upon this expertise to solve problems, achieve expert consensus, and to aid planning for more effective use of ergonomics information.

What specific services will CSERIAC provide?

Several symposia, workshops and short courses will be sponsored by CSERIAC each year. White paper panels and special study groups will be established as requested. CSERIAC will provide a range of information services including topical reviews, abstracts, annotated bibliographies, special analysis reports, computer-based models, design support software, methodological assistance, "Current Awareness Bulletins," and a peer-reviewed research journal.

Is CSERIAC planning to build a "meta-database" to take over the functions of existing individual databases?

Absolutely not. The purpose of CSERIAC is information analysis rather than archival storage. Hence, CSERIAC's goal is not to replace or duplicate existing databases but to provide a gateway to them. A gateway provides a single access point as well as help in selecting the most appropriate information sources to answer user questions. In addition to computer-based information sources, CSERIAC will facilitate contact and interaction with leading experts on crew system ergonomics.

Who may use CSERIAC's services?

CSERIAC will primarily support the Department of Defense and other government organizations, as well as U.S. Government contractors. It will also be available to other types of users: academic, corporate, and government; at both the domestic and international levels to the extent practical within DoD security guidelines and DoD policy regarding the handling of information on military critical technologies.

Does this mean all Military Services may request CSERIAC assistance immediately?

Yes, requests are welcome from all Military Services as well as NASA. During the next year CSERIAC will become fully operational and develop the capablity to offer its complete menu of services to all qualified users. To ensure that CSERIAC maintains a cosmopolitan rather than a parochial support role, its policy is governed by a Steering Committee consisting of representatives from the Army (Human Engineering Laboratory; Army Research Institute), the Navy (Naval Air Development Center), the Air Force (Armstrong Aerospace Medical Research Laboratory; Aeronautical Systems Division), and NASA (Johnson Space Center).

Where is CSERIAC located?

CSERIAC is hosted by the Armstrong Aerospace Medical Research Laboratory (AAMRL) and is located in government furnished offices at AAMRL on Wright-Patterson Air Force Base near Dayton, Ohio.

How should I contact CSERIAC?

Interaction with CSERIAC can be by mail, telephone, electronically (computer-based), or in person. Users are encouraged to request information services directly from CSERIAC when information analyses, evaluations, or judgments are needed. Special tasks will require approval by the CSERIAC project manager. In general, services will be on a cost-recovery basis. Until CSERIAC is operational in Summer 1988, you should contact the Program Manager Lt. Col. John Edwards or Technical Director Dr. Kenneth Boff at the Armstrong Aerospace Medical Research Laboratory's Human Engineering Division (AAMRL/HEX), Wright-Patterson Air Force Base, OH 45433-6573. To subscribe to "Current Awareness Bulle~~tins~~" and newsletters, send your request on your organization's letterhead to the Program Manager.

VISUAL-VESTIBULAR INTERACTION IN PILOT'S PERCEPTION OF AIRCRAFT OR SIMULATOR MOTION

88-4622-CP

R.J.A.W. Hosman and J.C. van der Vaart
Delft University of Technology
Faculty of Aerospace Engineering
Delft, The Netherlands

Abstract

The importance of man's vestibular organs in perceiving cockpit motion in an aircraft or a simulator is nowadays hardly questioned, as witnessed by the present widespread use of six-degrees-of-freedom motion systems for flight simulators. Still more advantage could be gained from the use of moving base simulators.

In order to illustrate this, the paper reviews research on control behaviour and performance of subjects in target following and disturbance tasks. By using results of work by the authors and by others, the importance of peripheral visual and vestibular motion perception in tasks that require inner-loop stabilization, is emphasized. Results of stimulus response experiments, especially designed to gather insight in central and peripheral visual and vestibular perception of motion are summarized and used to explain findings of tracking experiments.

It is concluded that peripheral visual and cockpit motion cues are of paramount importance in actual or simulated manual aircraft control and that, in simulation, the compensation for simulator motion system dynamics, computing time-delays and motion control laws deserve much more attention.

1. INTRODUCTION

The influence of motion cues on pilot's control behaviour in manual flight control and therefore also in flight simulation are considered much more important than some twenty years ago. As a consequence present day training flight simulators are equiped with six degree of freedom motion systems and the requirements on motion simulation for Phase I, II and III simulators are laid down in the regulations. Due to the subsequent development of visual display systems flight simulation has become a well matured technique with wide applications in pilot training and aerospace research and development.

However, flight simulation in itself will always have certain limitations, mainly in the field of time delays and motion simulation. Because of these limitations manual control of the simulated aircraft is considered a more demanding task then flying of the real aircraft. The present-day shortcomings of flight simulation should not be overemphasized but the authors feel that some further improvement is possible if more is known about the process of perceiving (simulated) aircraft motions.

In manual flight attitude stabilization is the inner loop of aircraft control. Since aircraft are higher order systems, it follows from elementary principles of control theory that for attitude stabilization not only attitude but also angular rate information should be available to the pilot to establish stability.

Experimental research indicate that in many situations, inner loop attitude stabilization may, after some training and at the cost of some mental effort, be attained by using central (or foveal) visual information only, for instance from flight instruments. However, experiments also indicate that the information mediated by peripheral vision and by the vestibular system is much more appropriate for inner loop stabilization. By these modalities, man perceives much more directly the rate of change - first time derivative - of aircraft motion variables.

A long series experiments performed at the Faculty of Aerospace Engineering of the Delft University of Technology yielded a better understanding of the contribution of the visual- and vestibular system to the pilot's perception of aircraft rotary motions. It is evident now that improvements in the motion perception process and the consequential improvements in tracking performance due to peripheral visual and vestibular motion cues results mainly from a faster perception process.

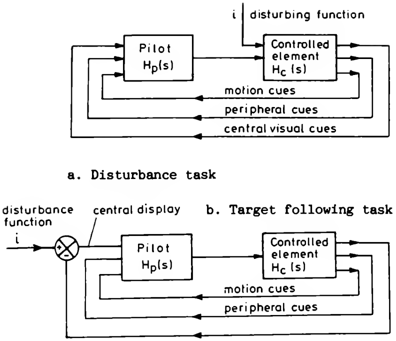
The research summarized in Chapter 2 is concerned with control behaviour and performance of subjects in target following and disturbance compensation tracking tasks. These tasks mainly require short term stabilization and error compensation, the disturbance compensation in fact is a pure stabilization task. In these experiments peripheral field displays and cockpit motion improved tracking performance and changed dynamic behaviour of the pilots if compared to the no-motion case with flight instruments only.

Chapters 3 and 4 are devoted to experiments more specifically aimed at the perception process in these control tasks, i.e. perception of attitude and angular rate by vision (Chapter 3) and perception of aircraft cockpit motion by way of the vestibular system (Chapter 4).

Chapter 5 summarizes the results of these and related experiments by others. Implications of the experimental findings for human vehicle control behaviour and flight simulation are indicated.

2.The influence of visual and vestibular motion perception on man-vehicle control.

A large number of investigations on the influence of motion perception on man-vehicle control were published during the last twenty years [6, 11, 12, 13, 15, 17, 19]. Most of these feature the use of moving base aircraft or car simulators. The amount of experimental data is impressive but results are difficult to compare, firstly due to a wide variety in simulated vehicle dynamics and simulator moving base dynamics, next, due to differences in the type of motion to be controlled. Some experiments are on lineair motion, others on rotational motion.



a. Disturbance task

b. Target following task

2.1 Block diagrams of the human controller and controlled element.
a. Disturbance task: all cues closely related.
b. Target following task: error displayed on central display not directly related to other cues.

In addition, two different types of tracking tasks may be distinguished. It is well known that in disturbance compensation tasks and target following tasks, see Fig. 2.1, cockpit motion has a completely different effect on subject's control behaviour[6, 12], although performance improves in either case.

Apart from these there may be differences in the visual displays and in the error scores to be minimized by subjects. Moreover subjects may range from naïve non-pilots in one experiment to experienced professional pilots in another.

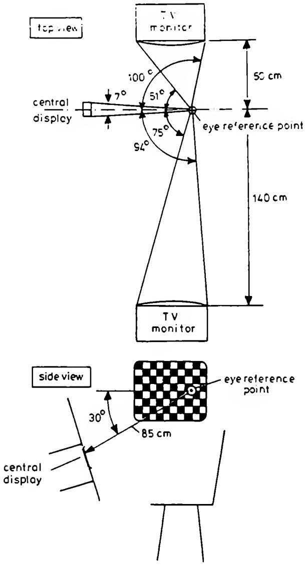
In spite of these mayor and minor differences in experimental apparatus and set-up, it is now well established that vestibular motion cues and peripheral visual cues have a beneficial effect on subjet's performance (more accurate tracking) and that they do change subject's control behaviour.

The influence of motion perception on tracking performance and control behaviour will be illustrated by results of roll tracking tasks only, since these tasks are most extensively documented. In these tasks, there are three major sources of information to the subject for perceiving aircraft roll motion.

- central visual information on the artificial horizon (central display)
- peripheral visual information from the outside world and
- cockpit roll motion as perceived by the vestibular system.

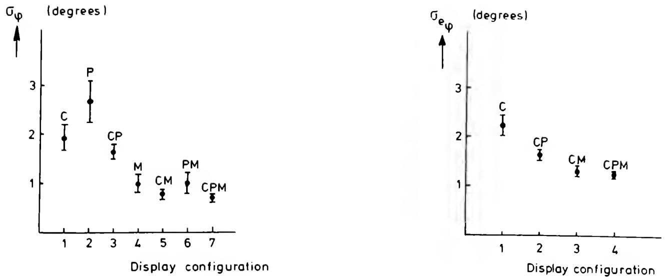
Examples of experiments featuring one or more of these sources of motion information can be found in the references[6, 11, 12, 13, 15, 17, 19]. Peripheral visual stimuli were used in several research programs[6, 15]. They were presented to subjects by vertically moving checkerboard or horizontal stripes patterns on two TV monitors, see Fig. 2.2. In some cases cockpit motion is "displayed" to subjects with a certain time-delay due to the dynamic characteristics of the simulator motion system. For compensatory tracking experiments, it is essential that motion cues presented by the visual displays and the motion system are accurately synchronized.

Before considering a number of results into more detail, it may be worthwhile to mention that a general conclusion to be drawn from all experiments is, that peripheral field displays and cockpit motion have a similar influence on tracking performance and control behaviour when added to a central display. This holds for disturbance tasks and for following tasks alike. In both cases changes in performance and subject's dynamic behaviour are, as a rule, larger due to cockpit motion than due to peripheral visual cues.



2.2 Position of central and peripheral displays relative to the subject's eye reference point.

For a more detailed analysis, the results of compensatory tracking tasks by the present authors[6] are now considered. A disturbance task and a target following task were done with a double integrator (K/s^2) as the controlled element. Disturbance tasks were run with all (7) combinations of the central display (C), peripheral displays (P) and cockpit motion (M), target following tasks were, by necessity, only run with combinations of displays featuring the central display (C, CP, CM, CPM). Since simulator and controlled element dynamics, visual displays and the general experimental set-up and procedures remained unchanged in these experiments, they offer a sound basis of comparison for the effect of several motion cues for the two types of control tasks.



a. Disturbance task

b. Target following task

2.3 Tracking error standard deviation as a function of display configuration for disturbance and target following task.

Fig. 2.3 shows the tracking error standard deviation as a function of display condition. The trend of these data corresponds very well with performance data of others.
In order to express measured control behaviour by a small number of parameters, the quasi lineair human operator model by McRuer [14] was used to match the data:

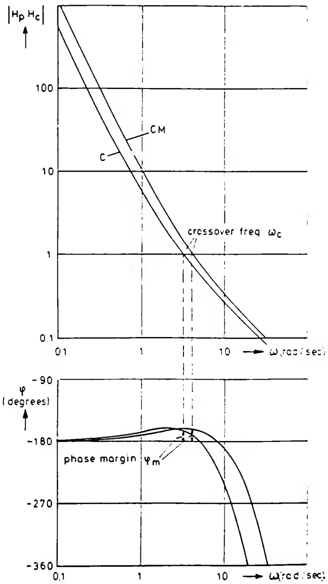
$$H_p(s) = K_p \cdot \frac{1 + T_L s}{1 + T_I s} \cdot e^{-\tau_e s} \qquad (2.1)$$

In eq. (2.1) K_p is the controller gain and T_I and T_L are human operator lag and lead terms respectively. The effective time-delay τ_e accounts for all time delays in perception, mental processing and control output generation. In addition, two more parameters i.e. the crossover frequency ω_c and phase margin Φ_m were calculated from the measured data. Crossover frequency is the frequency where the gain of the open loop frequency response $H_p(\omega).H_c(\omega)$ crosses unity, see Fig. 2.4. Phase margin is defined as the phase angle $\Phi(\omega_c)$ of $H_p(\omega).H_c(\omega)$ at ω_c, augmented with 180°:

$$\Phi_m = 180 + \Phi_{H_p . H_c}(\omega_c) \qquad (2.2)$$

A high crossover frequency is an indication of a high controller gain over a large frequency range, which is desirable for accurate disturbance compensation. However, there is a limit on the magnitude of controller gain, set by the requirement for closed loop stability. Phase margin is a measure for the stability of the man machine system. After training, well motivated subjects choose as high a controller gain as they feel to be compatible with a comfortable phase margin. For a detailed analysis, the reader is referred to Sheridan and Ferell [16].

Addition of the peripheral visual displays and/or cockpit motion affects the dynamic behaviour of the subjects differently in disturbance



2.4 Examples of the open loop describingfunction $H_p(\omega).H_c(\omega)$, the crossover frequency ω_c and the phase margin Φ_m.

tasks when compared to target following tasks, see Fig. 2.5 and [6, 12].

In disturbance tasks, effective time delay τ_e is seen to decrease. This enables subjects to increase the gain K_p, thereby increasing the crossover frequency ω_c, while they appear to be able to maintain an approximately constant phase margin ϕ_m, see Fig. 2.5 and Table 2.1.

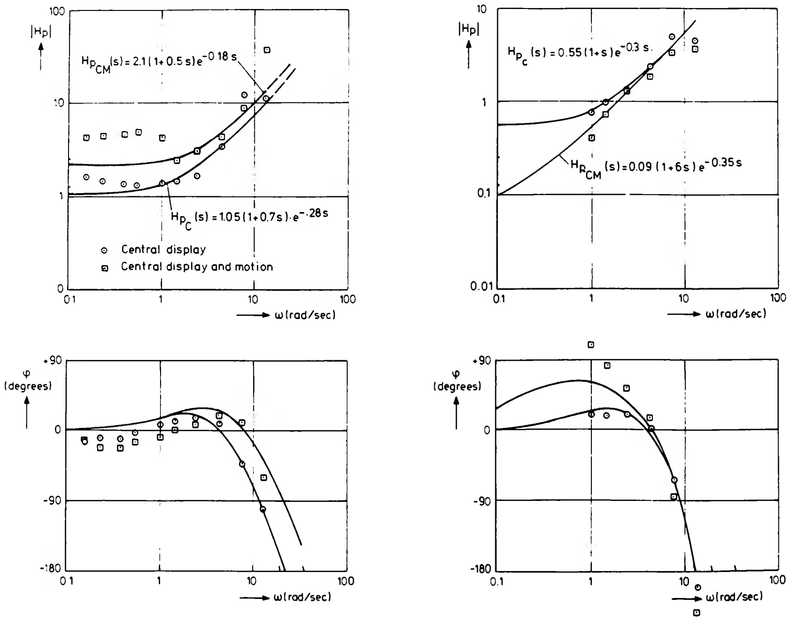
In the target following task the subjects use the additional information from the peripheral field displays and cockpit motion quite differently. Here subjects use the additional rate information to increase the time lead constant T_L while they decrease the static gain K_p. The result is an increase in phase lead in the low frequency range (ω < 3 rad/sec), see Fig. 2.5. Control bandwidth, as characterized by ω_c is approximately constant, but phase margins increase, see Table 2.1.

The different effect of peripheral field displays and cockpit motion on control behaviour in target following tasks compared with disturbance tasks is also reported in [6, 12]. For a detailed analysis the reader is referred to Van der Vaart and Hosman [18].

Table 2.1.: Crossover frequency, phase margin, quasi-lineair pilot model parameters and tracking performance from the tracking experiment[6].

Disturbance task.

Display con-dition	ω_c	ϕ_m	K_p	T_L	T_I	τ_e	$\sigma_{\ddot{\phi}}$
	rad/sec	degree		sec	sec	sec	degree
C	3.18	14	1.3	0.53	0.1	0.24	1.94
P	3.16	15	1.3	0.50	0.1	0.23	2.67
CP	3.55	15	1.3	0.55	0.1	0.24	1.64
M	4.64	21	2.7	0.30	0.1	0.13	0.99
CM	4.76	19	2.6	0.30	0.1	0.14	0.79
PM	4.89	19	2.6	0.30	0.1	0.13	1.00
CPM	5.03	18	2.6	0.30	0.1	0.14	0.70
			Following task.				
C	2.26	15	0.55	1.	0.1	0.35	2.23
CP	2.24	35	0.15	4.	0.1	0.35	1.63
CM	2.23	39	0.09	6.	0.1	0.35	1.32
CPM	1.66	56	0.08	6.	0.1	0.35	1.26



a. Disturbance task

b. Target following task

2.5 Examples of the gain and phase of the measured human controller describingfunctions in the disturbance and target following task.

Work by others on the influence of additional rate information as supplied by peripheral vision and cockpit motion [6, 11, 12, 13, 15, 17, 19] shows at least qualitatively similar effects. Quantitatively, however, changes in performance and control behaviour differ considerably among experiments. It will be shown in Chapter 5 that some of these differences can be explained by differences in the dynamics of the controlled systems.

3. The perception of vehicle motion from central and peripheral field displays.

The following two chapters summarize findings of stimulus-response type experiments in which the displays and simulator were exactly the same as those of the tracking experiments of Chapter 2. These tests were especially designed to gather data on the visual perception process involved in the tracking experiments reviewed in Chapter 2.

Experiment

Speed and accuracy of visual motion perception were investigated by presenting discrete, constant values of roll attitude and roll rate. Roll attitude was presented on the central artificial horizon display (denoted by C). In roll rate experiments, stimuli were either displayed on the artificial horizon (C), on the peripheral field displays (P) or simultaneously on both display types (CP). Since the peripheral field displays have a continuous checkerboard pattern, they do not yield any cues on roll attitude.

After observing a single stimulus during a preset exposition time, subjects were required to make an estimate of stimulus magnitude (roll angle in attitude estimation, angular rate in rate estimation experiments).

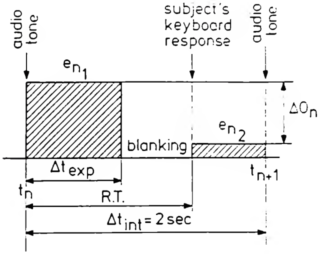
Stimulus magnitudes ranged from minus to plus 30 degrees with increments of 3 degrees for roll angle estimation. For roll rate experiments the range of the stimuli was ± 25 deg/sec, the increments being 2.5 deg/sec. On the keyboard a single key corresponded to the increments of 3 degrees of roll angle or 2.5 deg/sec of roll rate. The keyboard included 10 keys for clockwise rotations, 10 keys for counter clockwise rotation and a zero key, see Fig. 3.1.

3.1 Digital keyboard.

An experimental run contained 100 stimuli which were presented at fixed intervals, stimulus magnitudes were presented in random order and stimulus presentation was preceded by an audio tone in the subject's head phone, see Fig. 3.2. Immediately after a keyboard response, the remaining error (error roll angle or error roll rate depending on the experimental task) was shown on the displays, giving subjects immediate knowledge of the result. A correct response made the displayed angle return to zero in the roll attitude perception task. In the roll rate experiments a correct answer would "freeze" the displays.

Stimulus exposition time could be varied and was set at a constant value prior to each run. During a run perception errors and response times RT were recorded. After each run mean and standard deviations of these variables and a score parameter $S_c = \sigma^2_{e_{n_2}} / \sigma^2_{e_{n_1}}$ were computed.

Three subjects, university staff members and qualified jet transport pilots, volunteered in the experiment. Their instructions were firstly to respond as accurately and secondly as quickly as possible to the presented stimuli. All experimental conditions (5 for roll attitude and 15 for roll rate) were replicated 5 times for each subject. For a full description of the experiment the reader is referred to [7, 8].

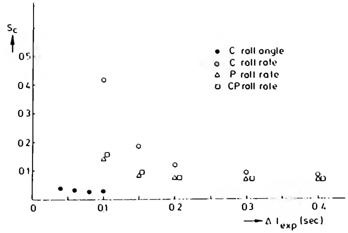


3.2 Sequence of one interval of a test run.

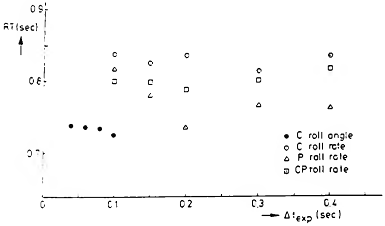
Results

In Fig. 3.3, it can be seen that if compared by the error score as defined above, roll attitude is perceived more accurately and within a much shorter exposure time than roll rate. In addition it is evident that a given accuracy for roll rate perception can be obtained in a shorter exposure time from the peripheral displays than from the central display.

Response times as a function of exposure time, are shown in Fig. 3.4. For the central display only, roll attitude response time is around 0.1 sec faster than roll rate response time. Roll rate perception from the peripheral displays shows a response time 0.06 sec shorter than the for the case of central display only. Roll rate perception from the central and peripheral displays combined shows scores approximately equal to the ones from peripheral displays although with slightly longer response times.



3.3 Score parameter S_c for roll attitude and roll rate estimation as a function of exposure time and display configuration.



3.4 Response time to roll attitude and roll rate stimuli as a function of exposure time and display configuration.

4. Vestibular motion perception.

In the Introduction, the important role of man's vestibular organ in inner loop attitude stabilization tasks was already mentioned. The present chapter summarizes the main dynamic characteristics of the vestibular system as far as its function for angular motion perception is concerned. The second part of this chapter reviews two stimulus response experiments which included cockpit motion.

Strictly speaking, the only physical input quantity to be sensed by the vestibular semi-circular canals, is angular acceleration. The output of this organ may be expressed in change in output neural firing rate, or more general as motion sensed by a human subject. Physically, the semi-circular canal organs have the dynamics of a heavily damped second order system. Input angular acceleration and motion as sensed by man would then be related by the transfer function:

$$H(s) = \frac{K}{(1+\tau_1 s)\ (1+\tau_2 s)} \qquad (4.1)$$

where K is a static gain, τ_1 = 10 sec and τ_2 = 0.005 sec for man, see van Egmond[2].

In the frequency range typical for human motion, the frequency response of the output of this system features a phase lag of approximately 90 degrees relative to the angular acceleration input. In frequency response, angular rate has exactly 90 degrees of phase lag relative to angular acceleration. It would be grossly incorrect to infer from this that the semi-circular canals can be considered as rate-sensors, but it may safely be said that the vestibular output due to most every day motions is correlated with input rate.

From neuro-physiological research, it is known that due to the differentiating action of certain sensory cells in the vestibular organs, an additional lead term should be introduced into the transfer function numerator:

$$H(s) = \frac{K\ .\ (1+\tau_3 s)}{(1+\tau_1 s)\ .\ (1+\tau_2 s)} \qquad (4.2)$$

where for man τ_3 = 0.11 sec.

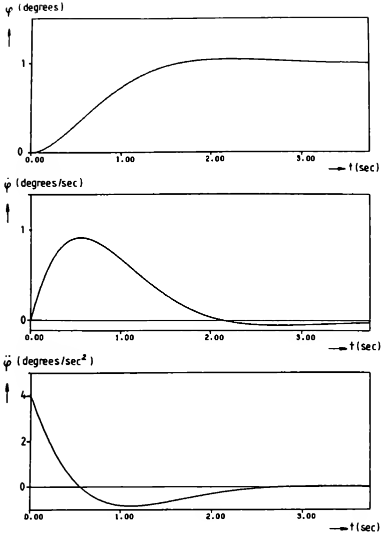
The lead time constant τ_3 was evaluated by Fernandez and Goldberg[3, 5] for the squirrel monkey. From measurements of thresholds for angular motion perception in a flight simulator a similar lead term τ_3 with the above mentioned value was determined by the authors[5].

The influence of the lead term on vestibular system model output is readily demonstrated by a numerical example. The step response of a second order system, simulating roll motion aircraft dynamics, see Fig. 4.1, was taken as the input to the semicircular canal dynamics according to equation (4.2). It can be seen from Fig. 4.2 that the vestibular output is advanced in time relative to the rate signal by 150 msec, suggesting an important advantage of vestibular motion perception when compared to visual motion perception.

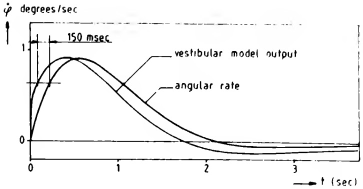
Experiment.

Second order system step responses as the one shown in Fig. 4.1 were used as stimuli in a stimulus response experiment using the same Delft University flight simulator as in the tracking tasks reviewed in Chapter 2 with exactly the same visual displays as in the former experiments. Subjects were asked to estimate the final roll angle magnitude by pressing the appropriate key on the keyboard also mentioned in the preceding chapter. The experiment was designed to investigate the temporal advantages due to cockpit motion and peripheral displays. Therefore in case of the display condition comprised the central display, subjects should be prevented from waiting till the final value of the step was reached. Subjects were trained to optimize their response between accuracy and speed such that for all display conditions the same subject strategy would be used.

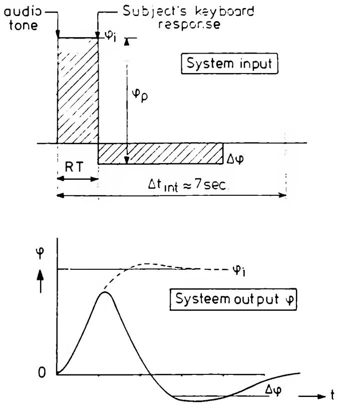
Just as in the other stimulus response experiments, response magnitude was subtracted from the input magnitude immediately after the subject's response. The second order system was, after subject's response, made to respond to this difference - or error - signal. After a transient motion, steady state response showed the error magnitude, see Fig. 4.3. In this way correct responses would return the simulator and the artificial horizon to zero roll angle. After an incorrect subject response the error was displayed



4.1 Attitude, rate and acceleration of a second order system step response



4.2 Angular rate output of a second order system step response and the calculated semi-circular canal output according to eq. (4.2).



4.3 Sequence of one interval of a test run.

during a few seconds, after which the roll angle was reset to zero before the next presentation.

Due to simulator limitations, step inputs were limited to ± 12 degrees, with increments of 2 degrees. This range was made to correspond with ± 6 keys on the keyboard, a single key corresponding with the stimulus increment of 2 degrees.

The natural frequency of the second order system was $\omega_0 = 2$ rad/sec, the damping ratio was $\zeta = 0.7$.

Stimulus magnitudes were presented in random order. During a single run, 65 stimuli were presented. The same seven display conditions as in the disturbance tracking tasks were used. Observation errors and response times were recorded. Two subjects, university staff members who also participated in the experiments described in Chapter 2 and 3, volunteered in the experiment. A full description of this experimental work is given in [9].

Results

Table 4.1 shows means and standard deviations of step response perception error and response times for 2 subjects and 5 replications, for the seven display conditions. From the mean response times, which are all under 1.3 sec, it can be seen that subjects are quite able to decide on the magnitude of the final value, well before the final, steady-state value is reached.

Peripheral displays shorten response time by 0.06 sec. and cockpit motion causes the response times to decrease by 0.21 sec. relative to the central display only condition. These changes were found to be significant ($p < 0.01$). Perception accuracy is only slightly improved by cockpit motion and appears to be hardly influenced by any of the other changes in display condition.

Table 4.1: Mean response time and perception error as a function of display condition.

Display condition	$\overline{RT}$	σ_{RT}	$\overline{\Delta\phi}$	$\sigma_{\Delta\phi}$
	sec	sec	degrees	degrees
C	1.163	0.162	0.15	1.34
P	1.098	0.174	0.32	1.39
CP	1.127	0.178	0.03	1.34
M	0.948	0.191	-0.06	1.19
CM	0.992	0.231	0.02	1.22
PM	0.905	0.173	0.16	1.26
CPM	0.940	0.212	0.09	1.25

Thus the experimental results comfirmed the above mentioned finding that the vestibular system needs a shorter time to perceive motion than the visual system. The outcome of the experiment did not exactly fit to the model based computations. Three reasons can be given for this:

- the paramaters of the semi-circular canal transfer function are estimates and not necessarily exactly correct
- the computed Δt of 150 msec is based on the difference between the vestibular model output and the stimulus rate ignoring the time-lag due to the visual angular rate perception
- cockpit motion stimuli may appear much stronger to subjects, which may influence response behaviour.

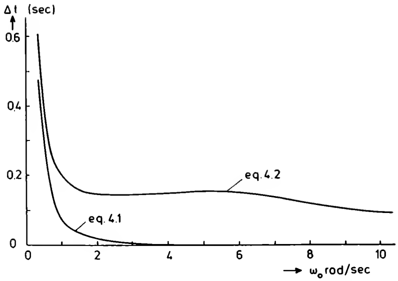
The influence of vestibular motion perception on the time needed to estimate the step magnitude depends on the dynamic characteristics of the system involved. This can be demonstrated by computing the time advance Δt See Fig. 4.4. Although there is a frequency range where this lead time is nearly constant, the lead of the vestibular output is seen to increase when the natural frequency ω_0 of the second order system decreases.

In order to evaluate if the influence of the contribution of the vestibular output on pilot's motion perception is as strong as suggested by Fig. 4.4, especially at low natural frequencies (ω_0 < 1 rad/sec), the experiment was extended. By using a second order system with different natural frequency to generate step response outputs, the response times of the subjects could be expected to change depending on display condition: central- or peripheral visual and/or vestibular stimulation. Thus it was expected that the relation between the time advance Δt and the natural frequency ω_0 could be evaluated experimentally.

From Fig. 4.4 it is clear that the natural frequency range of the second order system between 0.5 and 2 rad/sec is the most interesting due the increase of Δt with decreasing ω_0. For the details of these measurements see [10].

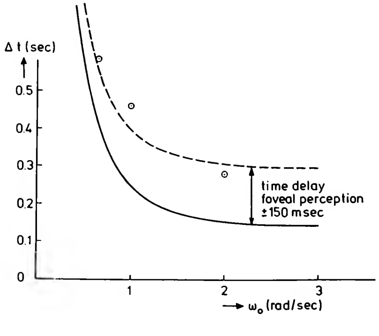
In Fig. 4.5 the time advance Δt as defined above is plotted together with the differences of the response times to the stimuli presented by the central display RT_C and cockpit motion RT_M for the three second order systems. Clearly the experimentally determined differences are larger than the model based time advance Δt. However, a sharp increase of $RT_C - RT_M$ with decreasing natural frequency ω_0 is found.

In van de Grind [4] it is shown that some time delay occurs in visual motion perception due to the fact that the stimulus has to pass the motion detector field on the retina before the associated neural circuit can generate an output. Time delays from 50 msec up to 2 sec depending on stimulus velocity are reported. This is confirmed by the stimulus response experiment, described in Chapter 3, where subjects had to estimate roll rate from the central display. It was shown that roll rate stimuli had to be exposed to the subjects for 100 to 200 msec before the roll rate could be perceived. In addition it has been found that the response time for roll rate perception is approximately 100 msec longer than for roll attitude perception. McRuer [14] described tracking experiments with first and second order dynamic characteristics. Based on these data it is assumed that the central visual system needs an additional 150 msec to perceive rate information. In Fig. 4.5 this time interval is added to the computed time advance Δt. The resulting line corresponds well with the values of $RT_C - RT_M$ found experimentally.



4.4 The time advance Δt of the semi-circular canal output relative to the roll rate signal as a function of the natural frequency ω_0 of the second order system.

Thus based on the model predictions and the experimental results it may be concluded that vestibular motion perception leads the central visual or foveal motion perception with an amount of time which depends on the characteristics of the motion signal to be perceived.



4.5 The response time difference $RT_C - RT_M$ compared to the time advance Δt and the time delay due to foveal motion perception.

5.Discussion and conclusions

From the results of the experiments described in Chapters 3 and 4, the basic differences in the perception characteristics of the visual and vestibular system for vehicle control can be summarized as follows.

- The time needed to perceive roll attitude from a central display is shorter than that to perceive its first time derivative, roll rate.
- Roll rate is perceived faster from the peripheral field displays than from the central display at equal accuracies.
- Due to the dynamic characteristics of the semicircular canals of the vestibular system the output is advanced in time relative the rate signal. This time advance increases with decreasing bandwidth of the motion signal to be perceived.

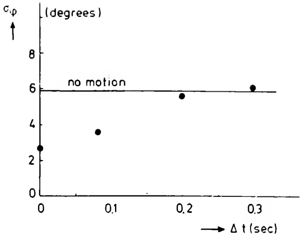
Some classic, well documented results from visual tracking tasks with a central display only can be explained by the above distinctions between roll angle and roll rate perception. It has long been known[14] that measured transfer functions for subjects controlling a single, respectively a double integrator, can well be approximated by:

$$H_{p_1}(s) = K_p \,.\, e^{-\tau_1 s} \qquad (5.1)$$

$$H_{p_2}(s) = K_p \,.\, s \,.\, e^{-\tau_2 s} \qquad (5.2)$$

where $\tau_1 = 0.24$ sec, $\tau_2 = 0.38$ sec.

Evidently, see eq. (5.1), a subject controlling a single integrator, will primarily observe the presented error signal itself, whereas a subject controlling a double integrator, see eq. (5.2), will observe the time derivative of the error signal as the principal variable. Bearing in mind the difference between continuous tracking and stimulus response type observation tasks, together with the approximative nature of eqs. (5.1) and (5.2), the difference in experimentally obtained time constants $\tau_2 - \tau_1 = 0.14$ sec can well be explained by the response times for rate perception, which were found to be about 0.10 sec longer than for attitude perception in Chapter 3.

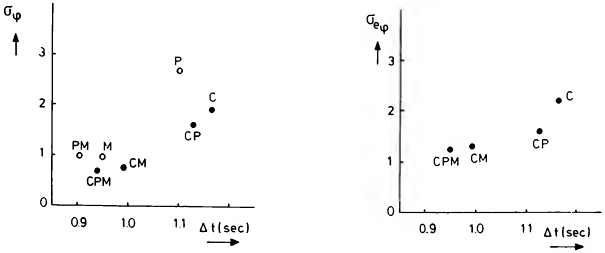


5.1 Root mean square of the roll angle in a roll tracking task as a function of simulator motion time delay Δt compared with the no motion performance, taken from Ref. 13.

Disturbance task

The shorter perception time of roll rate due to the addition of peripheral visual cues and vestibular cues provides the subject with the ability to decrease the time delay τ_e as expressed in eq. (3.1) when performing a disturbance tracking task. This smaller time delay is used to increase the crossover frequency ω_c at a constant value of the phase margin ϕ_m.

The decrease of the effective time delay τ_e as measured in disturbance tracking tasks due to the addition of peripheral visual cues (0.03 sec) and vestibular cues (0.10 sec), see Table 2.1, corresponds with the shorter response times in the sti-



a. Disturbance task

b. Target following task

5.2 Tracking performance in the disturbance and target following tracking task from Fig. 2.3, as a function of the response time RT from table 4.1, for the same display conditions. Open symbols: no central display.

mulus response experiments due to the same changes in display condition (peripheral cues 0.03 sec and vestibular cues 0,17 sec). See Chapter 3 and 4. Thus changes in time-delay in the tracking task can be attributed to changes in the perception process.

It stands to reason that if cockpit motion in a tracking task is delayed, the advantage of the early rate perception by the vestibular system will decline and eventually vanish. Hence the subject's control performance will decrease and the standard deviation of the controlled variable will increase. Results of an experiment on the influence of delayed cockpit motion on roll tracking performance by Levison [13] are shown in Fig. 5.1.

Taking the error standard deviations from Table 2.1 for the disturbance task and the response times of Table 4.1 for the corresponding display conditions, a plot similar to that of Fig. 5.1 can be made, see Fig. 5.2. Results of three conditions of the disturbance task (P, M and PM) deviate from the other conditions since, due to inaccurate roll attitude information to the subjects (no central display), tracking performance is not quite comparable with the other conditions.

Target following task

In the target following task subjects appear not to be able to speed up the perception process when peripheral visual or vestibular cues are added to the central visual cues. In contrast with the disturbance task, the peripheral visual and vestibular cues in target following tasks are not directly related to the first time derivative of the controlled variable, the roll error signal e_ϕ. See Fig. 2.1. In target following tasks subjects use the peripheral visual and vestibular feedback cues to increase the lead time constant T_L in the model of eq. (3.1), thereby increasing the phase margin ϕ_m while crossover frequency ω_c remains constant, see [18]. Again the influence of cockpit motion on changes in performance and dynamic behaviour is stronger than the influence of the peripheral field displays.

Concluding remarks

In this paper, the paramount importance of peripheral visual and especially vestibular information for inner-loop aircraft attitude stabilization was illustrated by results of tracking experiments. The emphasis on inner-loop control and vestibular feedback was not meant to imply that vision of the outside world scene is not important for certain other aspects of man-vehicle control such as orientation, navigation and trajectory control.

The summarized results of stimulus-response experiments indicate that the advantages from peripheral visual (outside world displays) and vestibular (cockpit motion) information are mainly due to the shorter perception times of these modalities when compared to central vision (primary flight display). The experimental results show that for larger aircraft (lower natural frequencies) cockpit motion provides the pilot with more time advance Δt than smaller aircraft.

In order to take full advantage of these systems, simulation time delays have to be minimized. Although the time advantage to be gained by cockpit motion is small when expressed in seconds, human performance and control behaviour are dramatically changed. It is therefore essential that the possible advantages to be obtained by costly flight simulator motion systems and visual displays are not nullified by delays caused by digital computer software and hardware [1].

6. References.

1. Baarspul, M., Hosman, R.J.A.W. and Vaart, J.C.van der. Some fundamentals of simulator cockpit motion generation. Proceedings of the International Conference on "Advances in Flight Simulation Visual and Motion Systems". Royal Aeronautical Society, London, April 29-May 1,1986.
2. Egmond, A.A.J. van, Groen, J.J. and Jonkees, L.B.W. The function of the vestibular organ. Practica-Oto-Laryngologica, Vol. XIV, Supplement 2, S.Karger, Basel, New York, 1952.
3. Fernandez, C. and Goldberg, J.M. Physiology of peripheral neurons innervating semi circular canals of the squirel monkey, III. Variations among units in their discharge properties, Journal of Neurophysiology, Vol. XXXIV, No. 4, 1971.
4. Grind, W.A. van de, Koenderink, J.J. and Doorn, A.J. van. The distribution of human motion detector properties in monocular visual field. Vision Research, Vol. 26, No 5, pp 797-810, 1986.
5. Hosman, R.J.A.W. and Vaart, J.C. van der. Vestibular models and thresholds of motion perception. Results of tests in a flight simulator. Delft University of Technology, Department of Aerospace Engineering, Report LR-265, 1978.
6. Hosman, R.J.A.W. and Vaart, J.C. van der. Effects of vestibular and visual motion perception on task performance. Acta Psychologica 48 (1981) 271-287.
7. Hosman, R.J.A.W. and Vaart, J.C. van der. Accuracy of visually perceived roll angle and roll rate using an artificial horizon and peripheral displays. Delft University of Technology, Department of Aerospace Engineering, Report LR-393, 1983.
8. Hosman, R.J.A.W. and Vaart, J.C. van der. Perception of roll rate from artificial horizon and peripheral displays. Delft University of Technology, Department of Aerospace Engineering, Report LR-403, 1983.
9. Hosman, R.J.A.W. and Vaart, J.C. van der. Accuracy of system step response roll magnitude estimation from central and peripheral visual displays and simulator cockpit motion. Proceedings of the Twentieth Annual Conference on Manual Control. NASA Conference Publication 2341, 1984.
10. Hosman, R.J.A.W., Roggekamp, R.P.G.M. and Vaart, J.C. van der. Contribution of vestibular system output to motion perception. Proceedings of the Twenty-third Annual Conference on Manual Control. Massachusetts Institute of Technology, Cambridge MA, June 1988, to be published.

11. Junker A.M. and Replogle, C.R. Motion effects on the human operator in a roll axis tracking task. Aviation, Space and Environmental Medicine, 46: 819-822, 1975.

12. Levison, W.H. A model for pilot use of roll axis motion cues in steady-state tracking tasks. Bolt Beranek and Newman Inc. Report No. 3808, Cambridge, Mass, 1978.

13. Levison, W.H., Lancroft, R.E. and Junker, A.M. Effects of simulation delays on performance and learning in a roll-axis task. Fifteenth Annual Conference on Manual Control. AFFDL TR 79-3134, 1979.

14. McRuer, D.T., Graham, D., Krendel, E.S. and Reisener, W. Human Pilot Dynamics in Compensatory Systems. Theory, Models, and Experiments with Controlled Element and Forcing Function Variations. AFFDL-TR-65-15, Wright Patterson AFB, OHIO, 1965.

15. Moriarty, D.T., Junker, A.M. and Price, D.R. Roll axis tracking improvement resulting from peripheral visionmotion cues. Twelfth Annual Conference on Manual Control, NASA TM-X-73, 170, 1973.

16. Sheridan, Th. B. and Ferell, W.R. Man-Machine Systems: Information, Control and Decision. Models of Human Performance. MIT Press, Cambridge Massachusetts, 1974.

17. Stapleford, R.L., Peters, R.A. and Alex, F.R. Experiments and a model for pilot dynamics with visual and motion inputs. NASA CR-1325, 1966.

18. Vaart, J.C. van der and R.J.A.W.Hosman. Compensatory tracking in disturbance tasks and target following tasks. The influence of cockpit motion on performance and control behaviour. Delft University of Technology, Faculty of Aerospace Engineering, Report LR-511, 1987.

19. Zacharias, G.L. and Young, L.R. Influence of combined visual and vestibular cues on human perception and control of horizontal rotation. Exp. Brain Res. (1981)41: 159-171.

MATCHING PILOT PERCEPTIONS OF REAL WORLD AND SIMULATED LIGHT SOURCES IN VISUAL FLIGHT SIMULATORS

88-4623-CP

Jack Tumblin
Manager of Advanced Research
IVEX Corporation 4357 J Park Drive
Norcross, GA 30093 (404) 564-1148

Abstract

Image generators for flight simulation often specify object brightnesses in display units, such as integer RGB values, so artistic judgment during color selection usually sets the relation between real-world and simulator brightnesses, here named 'brightness mapping'. As lighting and weather changes, such ad-hoc methods can cause inconsistencies in mappings which may distort a simulator pilot's perception of visibility, illumination, and distance. This paper gives a basic mathematical method and model for brightness mapping, and uses it to directly relate real-world brightnesses in units of optical power to display system units for an example simulator. The model is built from available human eyesight data, simulator display brightness and contrast performance, and five supported hypotheses about brightness perception.

Introduction

Existing simulator display systems have peak brightnesses of ones or tens of foot-lamberts, and achieve contrast ratios typically less than 100:1, yet routinely simulate real-world scenes with brightnesses anywhere from 0.00001 to 5000 foot-lamberts or more. By design or default there is a mapping process of real world brightnesses into display brightnesses to produce display images; this process is here named 'brightness mapping'.

Most image generators for flight simulation specify object brightnesses in display units, such as RGB values, so the artistic judgment of database builders during color selection usually sets the brightness mapping. As lighting and weather changes, such ad-hoc methods can cause inconsistencies in mappings which may distort a simulator pilot's perception of visibility, illumination, and distance.

Good flight simulation is a process of matching perceptions. Within the problem of matching all visual perceptions is found the need to match the perceived brightnesses of simulated items to the perceived brightnesses of these same items in the real world. Brightnesses here mean optical power per unit viewed area and it is understood that visible light can be represented as independent R, G, and B spectral components, so that 'brightness' as used in this paper may describe each component separately.

Though optical brightnesses meaningful in flight cover about ten orders of magnitude, they must be simulated on displays that cover no more than two of these. The human visual system is much better at judging differences in brightness than metering its value absolutely. Absolute brightness is apparently inferred from suggestion and the brightness changes noted within a display; hence a pilot's sense of overall brightness in the simulator might be set by direct suggestion and carefully chosen brightness mappings.

Such a brightness mapping is attempted here by using a systems design approach. The entire process of brightness simulation is first described in a block diagram, (Figure 1) including the unknown 'brightness mapping' process to be defined. Inputs, outputs, and equations for the processes of each block are defined where known, and then constraint conditions are used to link these equations. The equations are solved to express simulator display brightness directly in terms of brightnesses in the real world being simulated. To illustrate its usefulness, this result is applied to an 'actual', or example simulator model and an example real-world model, so that simulator attributes are expressed as equations of real-world attributes.

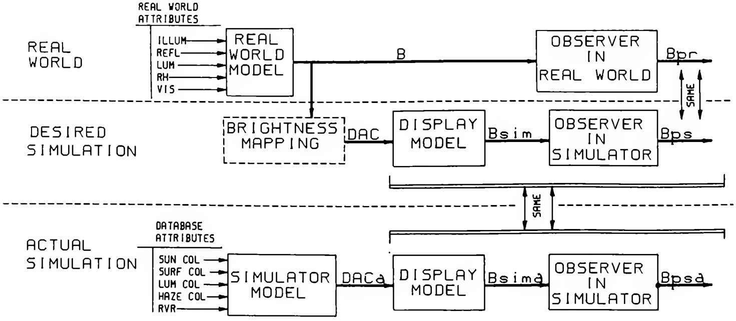


FIGURE 1: SIMULATION PROCESS MODEL

Brightness Simulation Model

Figure 1 is separated into three regions, where the top third represents the real world process we wish to simulate, the middle third the simulator performance we wish to achieve, and the bottom third is the actual simulator where the derived brightness mapping is applied. We will examine the model from top section downward. For simplicity, spatial frequency response effects throughout this paper will be mentioned, but neglected in calculations, and a steady state model is used; transient effects of eye adaptation are ignored.

The first block models the atmospheric and optical processes that produce a 2D brightness image B from the eyepoint, where B is in lamberts (1 lambert = 4/621 watts per square centimeter = 929.03 foot-lamberts). This is the optical power density per unit area radiating from the viewed surface, scaled by any losses that occur between the surface and the eye. B is a piecewise continuous 2D signal of unlimited spatial frequency content, representing a perfectly resolved image; real-world resolution-limiting processes, such as diffraction and eye focusing, can then be approximated as spatial filters on this signal. B is a function of 2 angular coordinates measured from the eyepoint, but B(theta,phi) will be written B for simplicity. The example real world brightness model used here is quite common;

$$B = vis*(Illum*Refl+Lum) + (1-vis)*Illum*RH \qquad (1)$$

where

Illum = brightness in lamberts of the illuminating source

Refl = reflectivity of the illuminated object, and $0 <= Refl <= 1$,

Lum = brightness in lamberts of self luminous surfaces such as lights,

RH = reflectivity of haze having infinite depth, where haze is the combined effects of water vapor and aerosols that impede vision.

vis = haze obscuration factor, $0 <= vis <= 1$, where vis=1 for no obscuration, and vis=0 for total obscuration by haze.

Directional and distance related effects may be included in determination of these terms, but are considered external to this model. 'Illum' source is often just the sun, with typical values of 11 lamberts at meridian and 0.04 lamberts near horizon [2].

The 'observer model' approximates human brightness perception, and is further described below and in figure 2. Its output is the real-world perceived brightness image Bpr, a filtered, modified version of B in which overall brightness is weakly sensed and affected by two factors of suggestibility.

The middle third of figure 1 shows the brightness simulation we wish to achieve. Real-world B is somehow obtained within the simulator, low-pass filtered by the resolution limits of the simulator (not shown), and transformed by the 'brightness mapping' process (to be determined), into display system input signal 'DAC'. DAC is normalized, such that $0 <= DAC <= 1$, where display emits minimum brightness when DAC=0 and maximum when DAC=1. Display system model here is linear with brightness, such that its output brightness Bsim is

$$Bsim = Acrt + DAC * Bcrt \qquad (2)$$

where

DAC = normalized display units, $0<= DAC <=1$

Bsim = display brightness in lamberts

Acrt = ambient or leakage brightness of the display system in lamberts

Bcrt = display gain in lamberts per DAC unit

Bsim is received by the 'observer in simulator' model, which is identical to the human observer model used in the top section of figure 1. Its output Bps is the perceived brightness in the simulator, which is equal to Bpr when suggestibility factors and brightness mapping function are correct.

The last third of the model represents an actual simulator using the brightness mapping model we wish to derive. The 'simulator model' block holds the visual system brightness calculations as done to determine DACa for the actual display. These calculations vary among simulators, so a nominal example is used; others are easily substituted.

$$DACa = (1-Vis\_sim)*(Sun*Surf+Lum\_sim) + (Vis\_sim)*Haze \qquad (3)$$

Some or all of these quantities may be directional, but such calculations are considered external to the model. All of these values are normalized, and so range between 0 and 1.

Sun	= normalized sun illumination (color)
Surf	= reflective surface color
Lum_sim	= luminous object color
Haze	= haze reflectance color
Vis_sim	= obscuration factor similar to vis in 'real world model'

The display model and observer model in the lower section of figure 1 are identical to the those in the middle section; during a good simulation their inputs and outputs will match as well.

Constraint conditions for good simulation are obvious; let perceived real-world brightness equal perceived simulator brightness, or Bpr = Bps. Further, let DAC in the desired simulator equal DAC in the actual simulator; DAC = DACa, and since both use the same display and observer model, Bsim = Bsima, and Bps = Bpsa. Given these constraints, we shall now

-solve for an equation for the block 'brightness mapping' block in the middle of figure 1,

-find an expression to directly relate real-world model attributes (fig.1, top) to the actual simulator database attributes (fig.1, bottom)

Before the constraints are applied, the detailed observer model of figure 2 will be developed and the underlying hypotheses presented.

Observer Model

The workings of a good observer model are not immediately obvious. While an unknown system is often modeled by describing its limitations, human vision is marvelously adaptive, covering its shortcomings so well that one tends to accept the perceived image as inherently correct. A modest survey of published vision studies indicates the eye's response to brightness is primarily logarithmic[4], and is affected by aiming, focusing, and dark adaptation mechanisms. In all four effects the eye adjusts to

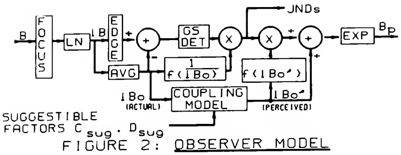


FIGURE 2: OBSERVER MODEL

follow its surroundings, and hence forms a servo system, but such transient effects are ignored in this model. The net effect of these mechanisms are approximated to find the observer model shown in figure 2, using published data and a few clearly marked hypotheses on visual perception.

The eye's lens system focuses an image of B on the detecting retina in the eye. Diffraction limiting and focusing error is modeled as a low-pass filter on B, named 'focus' in figure 2. Its 2D impulse response, sometimes called 'point-spread function' (PSF) in linear image processing [12], should include effects of intraocular scattering, where diffusing and diffracting materials in the eyeball's optical path cause glowing halos around high contrast areas. These effects are significant when viewing small intense lights, and can be modeled within the lens PSF [7]. As stated earlier, frequency response modeling will be neglected for brevity.

The retina's response to light varies greatly with position on the retina, in both sensitivity and in spatial frequency response (or less precisely, 'acuity') [1]. The eyeball has an agile aiming system which makes frequent jumpy scans, so that objects of interest within the brightness image B often overlay the high-resolution, color-detecting central (foveal) region of the retina. Unless 'dazzle' effects cause afterimages or trailing, scanning is not well perceived, and does not effect the perceived brightness or sharpness of the viewed scene. This suggests that the effects of retinal position are removed from Bp in the perception process, such that;

HYPOTHESIS #1: Normalized spatial frequency response in the perceived brightness image Bp, (a.k.a. acuity) is well modeled by a 2D spatial filter whose response is independent of position.

HYPOTHESIS #2 For viewed scenes without afterimages, the perceived brightness Bp of an object is not a function of position in the viewed scene.

The eye's response to brightness differences is approximately logarithmic, and most published measurement data uses logarithmic terms. To use this data, the natural logarithm is taken by the 'LN' block in figure 2, so that the detection processes that follow are written in ln (ln = natural logarithm) brightness form, and the results are restored to linear brightness by the 'EXP' block at the output. For brevity, the ln form of signals are written here with the prefix 'l', such that 'lBp' equals ln (Bp).

Perception of brightness is strongly influenced by edge detection mechanisms, so that illusions of brightness difference can be induced by strong edge effects, as demonstrated by the Hermann-Hering grid illusion of figure 3. Here, regions at the intersections of the white channels may appear dark. Though the neural processes that cause them are quite different, these edge effects have been shown by Fiorentini and Maffei (1973) [1,3] to be well modeled by a 2D spatial filter operating on perceived brightness, in which the response to brightness is reduced for upper and lower frequency components and boosted at middle frequencies. The block labeled 'EDGE' in figure 2 represents such a filter. This filter is significant in the observer model response to small intense lights, but will be neglected here for simplicity.

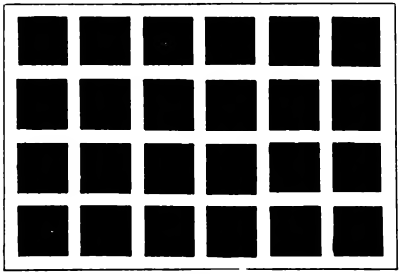


FIGURE 3

Human vision can discern brightnesses from about 1x10e-8 lamberts (starlit terrain) to 10 or more lamberts (snow at clear noon), but not all at once; there are limits to the contrast sensitivity of the eye. If two moderate- sized areas in a viewed scene are given contrast (Bmax/Bmin) of about 150/1, any additional contrast will be virtually undetectable, and the areas will be perceived as painfully bright white and perfectly dark black respectively. Thus, the eye as a detector of instantaneous contrast is overloaded beyond about 150/1, or whenever Bmax and Bmin are separated by more than 5 units on a natural log brightness scale [12]. To make such a detector usable over the entire gamut of 10/1x10e-8, or about 21 ln units, the eye adjusts its sensitivity, sliding these five units about the ln brightness scale over distances of up to 16 ln units. This adjustment is called 'adaptation', and adaptation assigns specific brightnesses to the contrast detection range of the eye.

Stated another way, the eye can be considered an ln brightness detector whose output is the perceived contrast, and only changes significantly over a range of 5 ln units, preceded by an attenuator, adjustable over 16 ln units, that is set by adaptation. If brightnesses are expressed in ln units, the attenuator is expressed as a subtractor, and the adaptation amount subtracted can be expressed as the natural log of a particular brightness, here named 'lBo'. If the contrast detector's active range is centered about zero ln units, then lBo will be the central ln brightness about which the eye has adapted. This analogy is used in the model of fig. 2, where the contrast detector is named 'GS detector' and adaptation occurs in the subtractor preceding it, and where lBo is determined from the focused image lB by the AVG block.

A simple model of GS detector is formed by assigning normalized units to its subjective contrast output. If the perception of 'painfully bright white' is assigned the value +.5 and 'perfectly dark black' is -.5, then the 'comfortable middle gray' value centered between these two extremes will be defined here as 0 if all other intervening values are distributed as equidistant by perceived contrast. This normalized, subjective gray scale number is here named 'GS', and the contrast response of the eye can then be directly related in these units.

The absolute range of the GS detector is already given as 5 ln units, other sources state viewer discomfort occurs for contrast beyond about 100/1 (4.6 ln units) [5,10], and yet contrasts as small as 5:1 (1.6 ln units) can form an image accepted as spanning the subjective range between black and white. Combined with inferences on the experiments of others (Heinemann 1955) [3,9] this suggests an S-shaped curve relates contrast to GS units.

The equation below models the GS detector as linear between -K and +K for the normalized ln brightness input 'lBn', and asymptotically approaches +.5 for lBn >K and -.5 for lBn<K, maintaining first derivative continuity at +/- K; see figure 4 for a plot of this function.

```
GS(lBn) = lBn / 5          for -K <= lBn <= K,    (4a)

     = .5 - (5*(.5-(K/5))**2)/(lBn + 2.5 -2*K)
       for lBn <' K                               (4b)

     = -.5+(5*(.5-(K/5))**2)/(-lBn + 2.5 -2*K)
       for lBn > K                                (4c)
```

Where
GS = normalized subjective gray scale value
lBn = natural log of normalized brightness
 K = ln of contrast breakpoint
(approximate suggested value; K=1.9)

Adaptation has been widely studied, and is the combined effects of several mechanisms, each operating at different rates. Adaptation can occur quickly over limited ranges, and different areas of the retina can adapt to different light levels, but these processes are not readily noticeable; therefore we will model adaptation with a single variable for the entire perceived image;

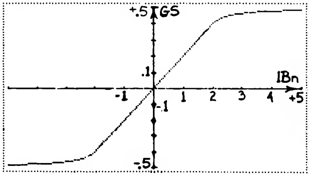


FIGURE 4

Hypothesis #3: The net effect of adaptation on the perceived image Bp can be well modeled as a function of one variable, here named overall scene brightness 'Bo'.

Hypothesis #4: Natural logarithm of Bo, lBo, is well approximated by the integral of lB over the entire viewed image.

Hypothesis #4 springs from the presumption that the eye adapts to maximize the amount of change and detail in the perceived contrast image GS. If such detail is symmetrically distributed about some ln brightness lBo, then the eye will probably adapt to place lBo at the center of the GS detector range.

The output of the GS detector expresses subjective contrast, but does not consider the eye's sensitivity to contrast itself, which has been shown to depend on adaptation. For example, about fifty shades are discernible between black and white in bright daylight, but only a few such shades can be seen by starlight. Adapted contrast sensitivity can be found from published measured data on 'visual threshold', which is the change in brightness required to make a test patch just barely noticeable against a uniform background. Naming background brightness Bbar and the difference between that and the just noticeably different test patch as deltaB, then visual threshold is the ratio deltaB over Bbar as plotted in figure 3 [5], and contrast sensitivity when adapted to Bbar is its inverse.

Since visual threshold is thus a quantitative measure of perception, it defines a crude measure of a single step in brightness perception itself. Some researchers [1,3,4] use assigned units to this measure, named 'JND' for 'Just Noticeable Difference(s)'. JNDs are especially useful here as a unit of perceived contrast for the eye as a contrast detector, so define

```
JNDs = (1/(deltaB/Bbar))* GS    @ Bbar = lBo
```

or in cleaner terms,

```
                JNDs = GS / f(lBo)                (5)
```

where the function [1/(deltaB/Bbar) @ Bbar=lBo] is renamed as '1/f(lBo)'. In figure 2 this signal is found by the block of the same name, and the multiplier that follows converts the GS detector output into units of Just Noticeable Differences, JNDs. A simple linear model for this function based on figure 5 is:

```
1/f(lBo) = 50      for lBo > ln(.003 lambert)     (6a)

           = 1/(3.3789x10E-3 * e**[-.3089*lBo])
                   for lBo < ln(.003 lambert)
```

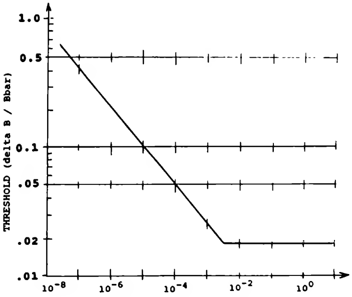


FIGURE 5 Bbar IN LAMBERTS

The relative contrasts in JNDs produced is probably the observer's primary source of information about the viewed scene, however the purpose of the model is to find the observer's perceived absolute brightness, Bp. Several references cite human vision as an excellent detector of contrast, but state that sensing of absolute brightness is weak and quite suggestible . For example, Ripps and Weale [1] suggest most visual information usable by the brain is sensed from relative contrast, so that perception of absolute brightness is of little importance, citing work by Ripps and Beare showing the eye is very poor or useless as a meter of absolute brightness. Despite this, when an opinion of absolute brightness is formed it is quite definite, and is likely to be a construction of sensation and suggestions from other cues, thus

---

Hypothesis #5: Perceptions of absolute brightness within a viewed scene are reconstructed from a combination of accurately sensed relative contrast (JNDs) and poorly sensed and suggestible adaptation factor lBo.

One's perceived value of lBo, is here called lBo', where the prime mark denotes its perceived value. The perceived value lBo' is produced in figure 2 by the COUPLING MODEL block, which represents the combined effects of all forms of suggestion on the sensing of lBo to form lBo'. While these are likely to be complex, they are modeled here as linearly related to lBo, where both the offset 'Csug' and the slope 'Dsug' are factors that are set directly by suggestion:

$$\mathrm{lBo'} = \mathrm{Csug} + \mathrm{lBo} * \mathrm{Dsug} \qquad (7)$$

If there is no error in perception due to suggestion, then Csug = 0 and Dsug = 1. Dsug differs from 1 when the perceived rate of change in lBo differs from the actual rate, and Csug is non-zero when a constant offset in lBo is erroneously perceived.

Such differences are common, for example, when driving deserted roads near dusk. While driving undisturbed, the eyes continually adapt to the failing light, that is, lBo is falling. The available contrast sensitivity, or 1/f(lBo), is enough so that sufficient contrast differences are perceivable (JNDs) for safe driving, but the rate of this adaptation is underestimated, or Dsug < 1. One drives on without headlights, convinced the road is brighter than it actually is, or Csug>0. The error is not corrected (Csug and Dsug are constant) until a light of known brightness, such as the unexpected headlights of another car, quickly changes ones opinion of the amount of remaining light and hence of Csug and Dsug.

The presumption that our coupling model is reasonable and that Csug and Dsug are settable and stable is necessary for the brightness mapping found in the next section.

The perceived brightness Bp is now found by the reverse process used to find JNDs from the focused brightness lB. The inverse of the 1/f(lBo) model, shown in figure 2 as f(lBo'), operates on lBo' to find the factor to convert JNDs back to normalized contrast. Then the inverse of the GS detector is treated as a null process, and the adaptation amount lBo' is added to the result. Finally, the logarithmic form is restored to linear by the 'EXP', resulting in the perceived brightness Bp. The model for these last steps is;

$$\mathrm{Bp} = e{*}{*}(\mathrm{JNDs} * f(\mathrm{lBo'}) + \mathrm{lBo'}) \qquad (8)$$

where

$$f(\mathrm{lBo'}) = .02 \quad \text{for } \mathrm{lBo'} > \ln(.003 \text{ lambert}) \qquad (9a)$$

$$= (3.3789\mathrm{x}10\mathrm{E}{-}3 * e{*}{*}[-.3089{*}\mathrm{lBo}]) \quad \text{for } \mathrm{lBo'} < \ln(.003 \text{ lambert}) \qquad (9b)$$

Collecting all terms and neglecting frequency domain effects and denoting functions by curly brackets such as f(x), the completed observer model is;

$$\mathrm{Bp} = e{*}{*}[\,(\,(f(\mathrm{Csug} + \mathrm{Dsug}{*}\mathrm{lBo}) \,/\, f(\mathrm{lBo})){*} \mathrm{GS}(\ln\,(\mathrm{B/Bo})\,)\,) + \mathrm{Csug} + \mathrm{Dsug}{*}\mathrm{lBo}\,] \qquad (10)$$

Solve for Brightness Mapping Model

Four constraint conditions are applied to link the models derived above and arrive at an expression for brightness mapping;

-1- Bpr=Bps, ; Perceived brightness in the real world equals perceived brightness in the simulator

-2- JNDr=JNDs; Just noticeable differences of perceived contrast in the real world matches just noticeable differences of perceived contrast in the simulator. This constraint is vital for proper visibility of lights through haze.

-3- Csugr = 0 and Dsugr = 1, hence lBorw = lBorw'; The observer in the real world has no errors in Bp due to suggestible factors in the coupling model

-4- The contrast (Bmax/Bmin) available from the display device is incapable of exceeding the linear range of the GS detector in the observer model.

Combined with the model equations given earlier, the brightness mapping relation is found algebraically. Constraints -2-,-3-, and -4- give

$$\mathrm{JNDs} = \mathrm{GS}(\ln(\mathrm{Brw/Borw})\,) \,/\, f(\mathrm{lBorw}) \qquad (11)$$
$$= \ln(\mathrm{Bsim/Bosim}) \,/\, f(\mathrm{lBosim})$$

and if Brw/Borw is within the linear range of GS, then

$$\mathrm{Bsim/Bosim} = (\mathrm{Brw/Borw}){*}{*}(f(\mathrm{lBosim})/f(\mathrm{lBorw})) \qquad (12)$$

where the suffix 'rw' denotes real world, and 'sim' denotes simulator. The expression shows a new term 'Bosim', which is the brightness in the simulator to which adaptation occurs. This second term means a second equation must be found to uniquely determine both Bsim and Bosim. Constraints -1- and -2- used with equation 8 gives

$$\mathrm{Bpr} = e{*}{*}[\mathrm{JNDrw}{*}f(\mathrm{lBorw}) + \mathrm{lBorw}] =$$
$$\mathrm{Bps} = e{*}{*}[\mathrm{JNDsim}{*}f(\mathrm{lBosim'}) + \mathrm{lBosim'}] \qquad (13)$$

which forces lBorw = lBosim' or more usefully,

$$\mathrm{lBorw} = \mathrm{Csug} + \mathrm{Dsug}{*}\mathrm{lBosim} \qquad (14)$$

Thus, equations 11 and 14 define a complete brightness mapping, including terms for pilot suggestibility set in the simulator, that can be directly applied to the simulator model.

Conclusions

While work thus far shows promise, more work is needed to complete this study. Once the example simulator model solution is given, the model should be expanded to account for frequency response characteristics, so the work will apply to more accurate display of airport lights. This paper should be considered as an attempt to move database modeling one step further from an art towards a descriptive science.

References

[1] "The Eye" Volume 2A, "Visual function in Man", Edited by Davson, Hugh, Part I 'On Human Vision' by Ripps, H., and Weale, R. A. (C) 1976 Academic Press, Inc., New York, NY

[2] "Handbook of Chemistry and Physics, 44th Student Edition" (C) 1963 The Chemical Rubber Publishing Co., Cleveland Ohio

[3] "Visual Perception" Cornsweet, Tom N., (C) 1970 Academic Press, Inc., New York, New York

[4] "Fundamentals of Visual Science" Rubin, Melvin L., & Walls, Gordon L., (C) 1969 Charles C Thomas Publishing, Springfield, Illinois

[5] "Modern Optical Engineering" Smith, Warren J., (C) 1966 McGraw-Hill

[6] "Military Standardization Handbook: Optical Design" MIL-Handbook-141, (C) 1962 Defense Supply Agency, Washington D.C.

[7] "Intraocular Light Scattering: Theory and Clinical Application" Miller, David and Benedek, George, (C) 1973 Charles C Thomas Publisher, Springfield Illinois

[8] "Vision Research Reports" presented before the Armed Forces - National Research Council Committee on Vision, NAS/NRC, Edited by Horne, E. Porter and Whitcomb, Milton A., 1960 National Academy of Sciences - National Research Council, Washington D.C.

[9] "Contrast Thresholds of the Human Eye" Blackwell, H. Richard, 'Journal of the Optical Society of American, vol. 36, no. 11, November 1946, pp624-643

[10] ACM Siggraph '85 'High Performance Image Generation Systems' short course; Peter Doenges "Overview of Computer Image Generation In Visual Simulation"

[11] "Fundamentals of Interactive Computer Graphics" James D Foley & Andries Van Dam, (C)1982 Addison- Wesley Publishing, Reading, Mass.

[12] "Digital Image Processing" Gonzalez, Rafael C., & Wintz, Paul, (C) 1977 Addison Wesley Publishing Co, Reading,Mass.

[13] "The Eye: Window to the World" Wertenbaker, Lael (C) 1984 Torstar books, Inc. New York,NY. figure 3 in this paper drawn after figure on page 133, credited to the Bettmann Archive

Biographical Sketch

Jack Tumblin is Manager of Advanced Research and one of the founders of IVEX Corporation, where he is involved in investigation, analysis, and design of real-time image generation products. Recent work has included IVEX's Raster Light Points product, in which some methods presented in this paper are applied, and analysis and design for the VDS-1000 daylight visual system's computational hardware. Early work at IVEX has included hardware for IVEX's first real-time terrain device, an automated 1" videotape single-frame recording facility for videodisc databases, and methods for digital timebase correction of videodisc signals. Before IVEX's formation, Mr. Tumblin worked in broadcast television engineering, and was a Research Engineer with the Georgia Tech Research Institute in Atlanta Georgia, where two real-time digital image generators were designed for investigations of optically guided missiles. Mr. Tumblin holds a Bachelor's Degree in Electrical Engineering from the Georgia Institute of Technology, and recently resumed work there toward a Master's Degree. He was granted one patent for work at Georgia Tech, and one for work at IVEX, with three others pending.

DIAGNOSTIC MEASUREMENT APPROACHES TO THE PROBLEM OF SICKNESS IN FLIGHT SIMULATORS

88-4624-CP

Lawrence J. Hettinger, Norman E. Lane, Robert S. Kennedy
Essex Corporation
1040 Woodcock Road, Orlando, Florida 32803

Abstract

Simulator sickness is characterized by a constellation of symptoms that resemble motion sickness, but are less severe and often involve visually-related disturbances less frequently observed in other forms of motion sickness. To date, most research on motion sickness has used a self-report index called the Motion Sickness Questionnaire to scale the severity of illness experienced by individuals in various inertial environments. The present paper describes the development of the Simulator Sickness Questionnaire and discusses its use as a diagnostic tool to identify engineering features of flight simulators which may lead to illness in users.

Introduction

The use of simulation in aviation training has steadily increased in recent years primarily due to factors associated with safety, availability, and cost effectiveness. Simulators permit practice on a variety of tasks, such as emergency procedures, which cannot be conducted safely, or well, in the aircraft, and can provide a variety of training options such as playback, freeze, and performance measurement and recording.

Technologically advanced simulators for training vehicular control skills are becoming increasingly commonplace. Unfortunately, an increased incidence of motion sickness-like symptomatology, referred to as "simulator sickness" and related perceptual aftereffects, has accompanied the increased sophistication of ground-based flight trainers.(1) Simulator sickness is characterized by a constellation of symptoms and signs, with slightly different patterns or profiles from "true" motion sickness. Individuals may experience this sickness during or after simulator training flights. The term "motion sickness" applies to the adverse consequences of exposure to certain environments characterized by physical displacement of an individual.(2) The incidence, time course, and symptom mix follow certain rules, some of which are known. Frequently, if the stimulus parameters are sufficiently specified, it may be possible to predict the resulting incidence of sickness.(3) It follows that, to the extent that the real system (i.e., the aircraft) produces motion sickness, a simulator which replicates the real environment is liable to induce the same responses. However, when a simulator produces symptomatology which is dissimilar from that which ordinarily occurs in the real system, then the training value of the simulator may be compromised. As pointed out by Casali(4), the term "motion sickness" cannot be used as a global description of sickness induced by simulators. Many simulators impart no physical motion at all, and yet sickness may still occur as a result of perceiving visual representations of motion.(5,6)

A variety of adverse symptoms and signs are known to occur during simulator usage, and often for some period of time thereafter. Pathognomic signs of vomiting and retching are rare, while other overt signs such as pallor, sweating, and salivation are more common.(1) Major reported symptoms include nausea, drowsiness, general discomfort, apathy, headache, stomach awareness, disorientation, fatigue, and incapacitation. Postural changes have also been observed directly after simulator flights.(7) Among the more serious problems presented by this syndrome are residual aftereffects(8,9), including illusory sensations of climbing and turning, perceived inversions of the visual field and disturbed motor control. While the symptoms of simulator sickness are superficially the same as those of conventional motion sickness, they tend to be less severe, of lower incidence, and appear to involve elements of visual and visuo-vestibular interaction not typically present in conditions which induce motion sickness.

Implications of Simulator Sickness

The implications of simulator sickness fall into three interrelated categories.

Safety and Health. Among the implications of simulator sickness for safety and health are the presence of locomotor ataxia, interference with higher-order motor control, physiological discomfort, and visual aftereffects or flashbacks. The aftereffects which are occasionally experienced following a simulator session are particularly like those observed in research with distorted or rearranged visual stimulation. The perceptual factors that trigger flashbacks, their duration, and their period of retention are all important issues for which there is meager information in the scientific literature. There is also little information in the simulator sickness literature concerning aftereffects other than that they occur and often outlast the stimulus for a considerable period of time. Their occurrence can, however, be presumed to pose a significant threat to pilots' safety immediately following use of the simulator, and in some cases mandatory grounding policies have been adopted following simulator training flights to guard against effects of disorientation in flight.(10)

Compromised Training. The occurrence of simulator sickness may result in poor training and the development of adverse attitudes towards the use of simulators as training devices. Symptomatology may interfere with learning in the simulator by various means, including distraction due to the presence of physiological disturbance. The plasticity of the human central nervous system makes it likely that trainees will eventually adapt to any unpleasant perceptual experiences in the simulator. However, the most critical problem is that behaviors learned in the simulator may not be similar to responses required in flight, potentially resulting in the acquisition of perceptual-motor behaviors inappropriate to flying.

Readiness and Operational Effectiveness. The occurrence of simulator sickness has implications for pilots' operational effectiveness, including down-time and the acquisition of habits inappropriate to control of operational systems. For example, to minimize pseudo-Coriolis effects(11) pilots may restrict head movements in the simulator. Pseudo-Coriolis effects occur when an individual viewing a rotating visual stimulus makes head movements outside the axis of rotation. This situation often produces feelings of discomfort similar to those experienced when head movements are made outside the plane of actual physical rotation.

Based on restrictions in force within the U.S. Navy and Marine Corps concerning flight activities subsequent to training in simulators with high sickness rates, pilots may be grounded for 12-24 hours following the simulated flight. If such restrictions were to occur throughout the Armed Forces, it has been estimated that the typical aviator's total operational availability could be reduced as much as 5%. Therefore, any reduction of simulator sickness may result in very high payoff in terms of improved pilot operational readiness.

Measurement of Simulator Sickness

Among the major problems involved in the identification of equipment and individual pilot factors that leads to the occurrence of simulator sickness is the lack of a psychometrically reliable and convenient means of assessing the degree of its occurrence and, in particular, the severity of the particular symptoms that are present. Virtually all the research to date on simulator sickness symptomatology has used a scoring procedure called the Motion Sickness Questionnaire (MSQ). The MSQ was developed more than 20 years ago for studying the causal mechanisms underlying motion sickness.(12,13) The MSQ arose from a need to assign numerical indices to the degree of motion sickness severity experienced by individuals in a variety of motion sickness-provocative environments, when symptoms terminated short of actual vomiting (emesis).

The MSQ has been applied by a number of investigators in several scoring formats. Variants of the MSQ were applied in studies of seasickness in naval vessels,(14,15) in aircraft during hurricanes,(16) under weightlessness, (12,17) in rotating environments,(13) and in a series of NASA studies predicting space adaptation syndrome.(18)

The MSQ consists of a list of (usually) 25 to 30 symptoms associated with or premonitory of motion sickness onset; an individual indicates the presence for each symptom and/or degree of severity at the time of MSQ administration. The symptoms are then converted to a score ranging from 0 to (variously) 5, 7, 16, etc., where 0 indicates no reported symptomatology at all and the highest score represents confirmed emesis. Intermediate scores are assigned according to a "configural" or "profile" method, based on a combination of how many symptoms are reported, which symptoms these are, and the degree of reported severity.

The objective of more recent MSQ developments has been to provide a scale that could reliably indicate the onset of motion sickness under conditions of stimulation materially less severe than those of earlier studies, which generally used a model of testing to an emesis endpoint. As such, the current MSQ scoring approach was the obvious method of choice in the studies of simulator sickness initiated in the early 1980's.(19) Symptoms of simulator sickness generally mimic those of motion sickness, but affect a lower proportion of the exposed population and are usually (except in extremely susceptible cases) of much less severity.

The different varieties of motion sickness (sea, air, space, simulator, etc.) have much in common. The symptomatologies are highly similar, but they are not isomorphic, and it is probable that the causal mechanisms underlying symptom production are likewise not identical. Simulator sickness itself can also produce different symptom clusters as a result of inter-simulator differences.(20-22) It is tempting to speculate that if different profiles of stimulation were presented in a simulator, it might be possible to infer which characteristics of the stimuli (e.g., pseudo-Coriolis, inertial motion spectrum, visual distortions, visual/motion lags, etc.) are producing the unwanted effects.

In order to make such inferences, it is necessary to draw much finer distinctions among symptom patterns than is possible with the single score produced by the conventional MSQ. The MSQ, although an adequate index of overall severity, lacks power for diagnosis, i.e., for isolating the specific mechanisms or subsystems that might be causing a particular simulator to have an abnormally high incidence of reported symptoms.

There are also some other differences between motion sickness and simulator-induced sickness which make the current MSQ an acceptable, but less than ideal, index of simulator problems. Because the stimuli that produce simulator sickness are similar to but of lower intensity than those that produce motion sickness, the symptoms on which the MSQ is based are reported less frequently and with lower severity for simulator exposures than for the motion stimulated samples on which the MSQ was calibrated. This has several implications.

First, some of the symptoms in the present MSQ are almost never reported under simulator exposure, or if noted, are indicated at a level that fails to exceed a "base level." Experience with the MSQ suggests that for each symptom, there is a base frequency with which that symptom will be checked by normal, healthy subjects who are not being exposed to any unusual stimulation at all. To the extent that in frequently checked symptoms fall within this base rate, those symptoms are not useful for scoring purposes, and, if scored, may actually be misleading as to the severity of overall symptomatology.

Second, some symptoms with relatively high base rates may be artifacts of differences between simulator and motion stimulation. Drowsiness, for example, is a key indicator of the onset of motion-induced sickness.(23) It is also, however, a frequently indicated symptom in simulators for which there are no other reported symptoms; the high incidence of drowsiness in that case may be an indicator of a relatively benign simulator environment, one in which the absence of noxious stimulation induces a simple "sleepiness" reaction quite different in meaning than the parasympathetically-induced "protective" reaction associated with drowsiness under powerful motion stimulation. Thus, some of the symptoms that are valid for scoring motion sickness are not necessarily appropriate for simulator sickness assessment.

These and other anomalies in the MSQ as a simulator sickness index emerged as a result of in-depth examination of symptom patterns for more than 1000 simulator exposures in ten different simulator(24) and led to the development of a "revised" MSQ, a "Simulator Sickness Questionnaire (SSQ).(25) This development had two major objectives: a) to provide a "cleaner," more appropriate and more "valid" index of overall simulator sickness severity as distinguished from motion sickness, and b) to provide "subscale" scores more diagnostic of the locus of simulator sickness in a particular simulator for which overall severity was shown to be a problem. By "sharpening" the measuring instrument in that way, the ability to do "differential diagnoses" on simulators may allow better identification and evaluation of engineering solutions to problems and lead ultimately to more precise specifications for simulator design and guidelines for simulator use. It will also be possible to produce a biocybernetic device which can caution the pilot or instructor about imminent simulator sickness distress before it reaches severe status.

Two other important uses emerge from the SSQ development. First, the configural scoring of the present MSQ makes it incovenient for machine administration and scoring. One of the major "fixes" recommended for current simulator problems is to track the incidence of simulator sickness symptomatology over time for a given simulator using a "quality control" model to detect shifts in calibration or other gradually emerging problems. This requires routine on-site data collection in a near "automated" mode; the scoring approaches for the SSQ make this monitoring and cumulative tracking relatively straightforward. Second, the subscale structure would allow for heuristic comparison of score profiles in different nauseogenic environments (space, sea, etc.).

Development of the SSQ Scoring Systems

General Approach

The data set used for development of the new SSQ systems is part of the Naval Training System Center simulator sickness data base. It consisted of 1119 pairs (pre and post exposure) of MSQs collected during on-site data collection studies performed at ten simulator sites during 1984. The MSQ version used in these studies contained the 28 symptoms shown in Table 1. Identification and description of the simulators studied and the general properties and composition of the data set are given in a separate document.(24) An extensive "fine-grained" analysis was performed on the symptom data, both pre and post simulator exposure, with the objectives of:

a) Determining which of the 28 symptoms showed systematic changes from pre-exsposure to post-exposure. Some symptoms were selected too infrequently to be of value as statistical indicators. Those with less than 1 percent frequency of occurrence were eliminated from further SSQ scoring analyses. Some symptoms showed either no changes in frequency/severity or actually decreased slightly from pre to post-exposure. These were also eliminated from subsequent analyses.

b) Determining which symptoms showed differential levels of frequency/severity across simulators. Since the ten simulators represented in the sample were known, by means other than the MSQ(7,8), to vary considerably in overall severity of simulator sickness problems, some variability in values across simulators was considered critical for inclusion of a symptom. Because inter-simulator differences were so large, conventional statistical tests were not very useful, and some judgment was exercised in deciding on the retention or elimination of symptoms on the variability criterion.

c) Determining symptoms which might give misleading indications. Some symptoms (e.g., boredom) had their highest frequency of occurrence in simulators which had little or no other indicated symptomatology, and were rarely seen in simulators which had high frequency/severity on most other symptoms. These were also eliminated from subsequent analyses.

Altogether, 12 of the 28 symptoms were eliminated from the SSQ for one or more of the above reasons. These are so identified in Table 1. It should be noted that some of these omitted symptom variables remain as important indicators of motion sickness, and are still for some circumstances potentially useful signs of simulator sickness; they do not, however, have the statistical properties that make them usable in quantitatively-derived scoring systems for simulator sickness. Vomiting, for example, is clearly an important sign of motion sickness and simulator sickness, but occurred only twice in

TABLE 1
Symptoms in MSQ and SSQ

MSQ Symptom	SSQ	
	Retained	Eliminated
General Discomfort	X	
Fatigue	X	
Boredom		X
Drowsiness		X
Headache	X	
Eye Strain	X	
Difficulty Focusing	X	
Increased Salivation	X	
Decreased Salivation		X
Sweating	X	
Nausea	X	
Difficulty Concentrating	X	
Depression		X
Fullness of Head	X	
Blurred Vision	X	
Dizziness (Eyes Open)	X	
Dizziness (Eyes Closed)	X	
Vertigo	X	
Visual Flashbacks		X
Faintness		X
Awareness of Breathing		X
Stomach Awareness	X	
Decreased Appetite		X
Increased Appetite		X
Desire to Move Bowels		X
Confusion		X
Burping	X	
Vomiting		X

approximately 1200 simulator exposures, too low a rate for correlations and other statistical data to be stable. Vomiting would thus be retained in the MSQ for studies of motion sickness, but is not scored in the SSQ described here.

It has been conventional in studies with the MSQ to use differences between pre and post scores as the main indicator of problem severity, i.e., a "deviation from baseline" measure. This can result at times in anomalies such as "negative changes," which can give dramatically incorrect interpretations when summary statistics are employed. Difference scores also present the usual problems associated with change scores such as chronically low reliability.(26) It became clear, however, in the course of the SSQ development that the pre-exposure MSQ contains very little useful information, provided that records from subjects who reported themselves as "other than healthy" are excluded from analysis. As part of MSQ administration, a pre-exposure checklist is employed in which respondents are asked if they are "sick" or in other than their "usual state of fitness." When respondents who gave positive answers to either of these two questions were dropped, there was very little variance remaining in the pre-exposure data. Inexplicably, this information is not obtained in all motion sickness and simulator sickness studies, although it is well-established that illness modifies motion sickness susceptibility thresholds.(12,27)

On the basis of the above, Lane and Kennedy (25) examined three related scoring systems which vary in terms of their complexity and ease of use. All three systems are based on factor analytic models of symptom subgroups within the questionnaire. As noted above, simulator sickness has multiple causes and produces a variety of different classes of symptoms. To the extent that simulators tend to produce one class of symptoms more than others, or individuals tend to respond to equivalent simulator exposure with symptoms of one class more than others, those tendencies will be apparent in the intercorrelations of reported symptom severity, and the resultant "clusters" of symptoms can be identified by factor analysis.

Two forms of factor analysis were used. The "conventional" method was a principal factors analysis, iterated until communalities stabilized, followed by normalized varimax rotation. This approach produced factors which, while theoretically orthogonal (independent), were correlated whenever all symptoms shared at least some variance in common, i.e., there was a "general" factor present on which all variables had "real" non-zero loadings. For purposes of diagnostic use of a scale, it may be desirable to have subscales which are as independent (cleanly a measure of a single component) as possible. To determine the presence and magnitude of the general factor, the hierarchical factor analysis method was used.(28) This method extends the analysis of the rotated factor matrix to extract a general factor (if there is one) and two or more group factors. (Group factors are those on which some subset of variables will have large loadings while other variables will have loadings near zero.) Group factors obtained in this way are typically "cleaner" (much less correlated) than those obtained from varimax rotation.

Analyses were conducted extracting 3-, 4-, 5-, and 6 factor solutions from the 16 symptom variables. Although all solutions were interpretable, the 3 factor solution was the most "sensible" and offered the most potential for SSQ scoring. Three distinct symptom clusters were found, tentatively labelled as: 1) Visuomotor (eyestrain, difficulty focusing, blurred vision, headache, etc.), 2) Disorientation (dizziness, vertigo), and 3) Nausea (nausea, stomach awareness, salivation, burping). Each of the three factors was used as the basis for each SSQ subscale.

The dimensionality of the symptom matrix is supported by factor analytic results using variants of the MSQ symptoms in related domains. Morrissey and Bittner(29), in a study of visual display unit (VDU) users, found symptoms reported after 3 hour sessions to be remarkably similar to those of motion sickness. Their analysis of symptoms produced 4 factors, 3 of which (dizziness, nausea, and blurred vision) correspond closely to those above. The fourth factor, general discomfort, may be a result of musculoskeletal or postural discomfort induced by the 3 hour session, or of the small sample size (N=18).

Bittner and Guignard(30) reported a 2 factor solution for motion sickness symptoms at sea using an 11 symptom questionnaire; reanalysis of their data suggests, however, that a 3 factor solution is equally plausible, and the three factors obtained are essentially coincident with minor variations to those of the present analysis. Bittner (1988, personal communication) also reports a 3 factor solution from an 11-symptom questionnaire administered to soldiers controlling a remotely piloted vehicle from a display inside a windowless van, with symptom patterns similar to those in the Morrissey and Bittner(29) VDU study, although the third factor may be an artifact of heat buildup in the vehicle.

All of the 2 to 4 factor solutions in the above studies are much like one another and much like those conducted by Lane and Kennedy.(25) There is invariably a visual factor and a nausea factor, and usually a factor with dizziness, disorientation, blurred vision and/or sweating. Discrepancies in factor patterns across these analyses are remarkably small given the differences in the stimulus domains and the symptom lists used, and strongly reinforce the appropriateness of the 3 factor solution as the basis for scale development.

The 3 factor solution suggests the existence of three (partially) independent symptom clusters, each reflecting the impact of simulator exposure on a different "target system" within the human. A given simulator may cause symptoms that fall into none, one or more, or all of these clusters, depending on the mechanism(s) by which the human is affected. This target system organization of symptoms has both theoretical and practical importance. It may eventually be useful in studying the physiological basis of the reported symptoms; it likewise simplifies the process of determining where and in what ways a simulator may be causing problems for the user.

The SSQ scoring system uses symptom weights based on the factor structures, and is calibrated to produce equivalent zero points and equivalent variability across the three scales and the total severity score. This facilitates comparisons among scales within a study and examination of results across simulators. The scoring system has been used in two simulators outside the calibration sample. In a reanalysis of data from a U.S. Army helicopter simulator reported by Gower, Lilienthal, Kennedy and Fowlkes(20), the SSQ scales showed power to differentiate the calibration clusters, along with score variability nearly identical to the calibration sample. In an evaluation of a new helmet mounted projection system (reference to tech report), the SSQ Visuomotor scale was elevated, indicating a potential problem with headache and eyestrain, while the more global MSQ score failed to indicate the presence of any significant symptomatology.

Discussion and Conclusions

Some summary statements follow from the foregoing:

The patterns of symptom presence and severity associated with simulator sickness are sufficiently different from those of motion sickness to justify the use of separate measuring systems tailored to quantification of those specific patterns.

The current Motion Sickness Questionnaire, despite some content inappropriate to simulator sickness-specific symptomatology, functions surprisingly well as an index of overall simulator sickness severity. Its principal deficiency is the global nature of the single MSQ score, which lacks power for differentiating among the probable cause(s) of a simulator problem.

In both the analysis performed by Lane and Kennedy(25) and related analyses, there seem to be at least three separate dimensions underlying motion sickness and related problems (simulator sickness, space adaptation syndrome, air sickness and maybe even mountain sickness, etc.). Each of these dimensions operates through a different "target" system in the human organism to produce "undesirable" symptoms. The importance of identifying and understanding these dimensions is that the mechanisms for amelioration and control may be different for each affected target system.

Three alternative versions for SSQ scoring, based on factor analytic models, were developed by Lane and Kennedy.(25) Two of the three provide both good indications of overall simulator sickness severity and reasonably powerful subscale scores for diagnostic purposes. The simplest of these, using unit weights on variables identified by varimax rotation, will likely be preferable for most applications. A deficiency of both these scoring systems is that subscales are more highly correlated than is optimal for diagnostic use. A third scoring system uses hierarchical factor rotation to produce subscales with much lower interdependence. The limited number of simulator sickness-relevant symptoms in their analysis, however, was not sufficient to properly define and anchor the subscales, and the reliability of the scales is at present too low to recommend the use of the third scoring system. An important recommendation of their work was that further development of hierarchically-based scaling be pursued. Given the improved factor definition possible with an expanded questionnaire, the more "independent" scales of the hierarchical system would likely be the method of choice for assessing simulator sickness symptomatology.

Since it appears that the 16 symptoms retained in the SSQ are sufficient but marginal for discrimination among simulators, that symptom list or one of equivalent size and diversity is suggested as an absolute minimum for future studies of simulator sickness, motion sickness, or related phenomena. To avoid problems of unreliability with derived scores, studies in new domains should give special attention to the size and content of symptomatology questionnaires or surveys.

Acknowledgments

Ideas contained herein are partly a result of the authors' work over the past eight years in projects supported by DoD and DoE which provided opportunities to collect and analyze significant amounts of data pertaining to motion sickness and simulator sickness. In particular, support was received from the U.S. Navy and Martin Marietta Energy Systems, Inc. The opinions of the authors are their own and do not reflect the official views of these sponsors.

References

1 Kennedy, R. S., Hettinger, L. J., & Lilienthal, M. G. (1988). Simulator sickness. In G.R. Crampton (Ed.), Motion and Space Sickness. Boca Raton, FL: CRC Press.

2 Money, K. E. (1970). Motion sickness. Psychological Review, 50, 1-39.

3 McCauley, M. E. & Kennedy, R. S. (1976). Recommended human exposure limits for very low frequency vibration (TP-76-36). Point Mugu, CA: Pacific Missile Test Center.

4 Casali, J. G. (1986). Vehicular simulation-induced sickness: Volume I: An overview (NTSC-TR-86-010). Orlando, FL: Naval Training Systems Center.

5 Hettinger, L. J., Nolan, M. D., Kennedy, R. S., Berbaum, K. S., Schnitzius, K. P., & Edinger, K. M. (1987). Visual display factors contributing to simulator sickness. In Proceedings of the Human Factors Society 31st Annual Meeting (pp. 497-501). New York, NY.

6 Parker, D. M. (1971). A psychophysiological test for motion sickness susceptibility. Journal of General Psychology, 85, 87-92.

7 Fowlkes, J. E., Kennedy, R. S., Lilienthal, M. G., & Dutton, B. (1988). Postural and psychomotor performance changes in Navy pilots following exposures to ground-based flight trainers (NAVTRASYSCEN TR-87-010). Orlando, FL: Naval Training Systems Center.

8 Baltzley, D. R., Gower, D. W., Kennedy, R. S., & Lilienthal, M. G. (1988). Delayed effects of simulator sickness: Incidence and implications. Paper presented at the 58th Annual Medical Association, New Orleans, LA.

9 Kellogg, R. S., Castore, C., & Coward, R. E. (1980). Psychophysiological effects of training in a full-mission simulator. Preprints of the 51st Annual Meeting of the Aerospace Medical Association (pp. 203-208). Anaheim, CA.

10 FITRON 124 (1981). F-14 WST 2F112/WAVS aircrew readjustment. U.S. Navy message from FITRON ONE TWO FOUR TO COMFITAEWWINGPAC, San Diego, CA.

11 Dichgans, J., & Brandt, T. (1973). Optokinetic motion sickness as pseudo-Coriolis effects induced by moving visual stimuli. Acta Otolaryngologica, 76, 339-345.

12 Kellogg, R. S., Kennedy, R. S., & Graybiel, A. (1965). Motion sickness symptomatology of labyrinthine defective and normal subjects during zero gravity maneuvers. Aerospace Medicine, 36, 315-318.

13 Kennedy, R. S., Tolhurst, G. C., & Graybiel, A. (1965). The effects of visual deprivation on adaptation to a rotating environment (NSAM 918). Pensacola, FL: Naval School of Aerospace Medicine.

14 Kennedy, R. S., Graybiel, R. C., McDonough, R. C., & Beckwith, Fr. D. (1968). Symptomatology under storm conditions in the North Atlantic in control subjects and in persons with bilateral labyrinthine defects. Acta Otolaryngologica, 66, 533-540.

15 Wiker, S. F., Kennedy, R. S., McCauley, M. E., & Pepper, R. L. (1979). Susceptibility to seasickness: Influence of hull design and steaming direction. Aviation, Space, and Environmental Medicine, 50, 1046-1051.

16 Kennedy, R. S., Moroney, W. F., Bale, R. M., Gregoire, H. G., & Smith, D. G. (1972). Comparative motion sickness symptomatology and performance decrements occasioned by hurricane penetrations in C-121, C-130, and P-3 Navy aircraft. Aerospace Medicine, 43(11), 1235-1239.

17 Lackner, J. R., & Graybiel, A. (1982). Rapid perceptual adaptation to high gravitoinertial force levels: Evidence for contest specific adaptation. Aviation, Space, and Environmental Medicine, 53, 766-769.

18 Reschke, M. F., Homick, J. L., Ryan, P., & Moseley, E. C. (1984). Prediction of the space adaptation syndrome. AGARD Conference Proceedings No. 372: Motion Sickness: Mechanisms, Prediction, Prevention, and Treatment (pp. 26-1-26-19). Neuilly-sur-Seine, France: AGARD.

19 Frank, L. H., Kennedy, R. S., Kellogg, R. S., & McCauley, M. E. (1983). Simulator sickness: Reaction to a transformed perceptual world. I. Scope of the problem (NAVTRAEQUIPCEN TN-65). Orlando, FL: Naval Training Equipment Center.

20 Gower, D. W., Lilienthal, M. G., Kennedy, R. S., & Fowlkes, J. E. (1987). Simulator sickness in U.S. Army and Navy fixed- and rotary-wing flight simulators. Proceedings of the AGARD Medical Panel Symposium on Motion Cues in Flight Simulation and Simulator Induced Sickness, Brussels, Belgium.

21 Kennedy, R. S., Berbaum, K. S., Allgood, G. O., Lane, N. E., Lilienthal, M. G., & Baltzley, D. R. (1987, September). Etiological significance of equipment features and pilot history in simulator sickness. Proceedings of the AGARD Medical Panel Symposium on Motion Cues in Flight Simulation and Simulator Induced Sickness, Brussels, Belgium.

22 Ungs, T. J. (1987). Simulator sickness: Evidence for long-term effects. Proceedings of the Human Factors Society (p. 505), Santa Monica, CA.

23 Graybiel, A., & Knepton, J. (1976). Sopite syndrome: A sometimes sole manifestation of motion sickness. Aviation, Space, and Environmental Medicine, 47, 873-882.

24 Kennedy, R. S., Lilienthal, M. G., Berbaum, K. S., Baltzley, D. R., & McCauley, M. E. (In Press). Simulator sickness in U.S. Navy flight simulators. Aviation, Space, and Environmental Medicine.

25 Lane, N. E., & Kennedy, R. S. (1988). A new method for quantifying simulator sickness: Development and application of the Simulator Sickness Questionnaire (SSQ) (Technical Report EOTR 88-7). Orlando, FL: Essex Corporation.

26 Cronbach, L. J., & Furby, L. (1970). How to measure change Or should we? Psychological Bulletin, 74, 68-70.

27 DeWit, G. (1957). Acquired sensitivity to seasickness after an influenza infection. Pract. Otorhinolaryngologica, 19, 579-586.

28 Wherry, R. J. (1984). Contributions to correlational analysis. Orlando, FL: Academic Press.

29 Morrissey, S. J., & Bittner, A. C., Jr. (1986). Vestibular, perceptual, and subjective changes with extended VDU use: A motion sickness syndrome? In W. Karkowski (Ed.), Trends in Ergonomics/Human Factors III. New York: North-Holland.

30 Bittner, A. C., & Guignard, J. C. (1988). Shipboard evaluation of motion sickness incidence. In F. Aghazadeh (Ed.), Trends in Ergonomics/Human Factors V. New York: North-Holland.

TIME DELAY COMPENSATION USING SUPPLEMENTARY CUES IN AIRCRAFT SIMULATOR SYSTEMS

Michael S. Merriken *
William V. Johnson
Jeffery D. Cress

Systems Research Laboratories, Inc.
A Division of Arvin/Calspan
Dayton, OH 45440

Gary E. Riccio
Armstrong Aerospace Medical Research Laboratory
Wright-Patterson Air Force Base, OH

Abstract

This study investigated the effects of providing real-world supplementary information to the visual and tactual modalities to reduce the deleterious effects of a delayed primary display on operator control performance. The supplementary visual and motion cues were presented at two different update rates: (1) at the same rate as the primary display and (2) at a rate 133 ms faster. The results indicate that the conditions with the faster updating secondary cues had better performance in altitude control than the conditions with the cues at the same rate as the delayed primary display. There were no significant effects for heading control. When compared to a control condition with no supplementary cue there were no statistical differences but the trend of the faster updating secondary cue conditions having better performance scores was maintained for both altitude and heading control.

Introduction

The interest in temporal fidelity research specific to aircraft simulators has increased as the use of simulators becomes more prevalent. One of these areas of research is delay compensation. Given that an aircraft simulator has some inherent transport delay, attempts have been made to reduce the effects of delay on operator control performance. Several linear compensation techniques have been proposed: single-interval lead and Taylor series expansions time domain compensation [1,2] and lead/lag frequency domain compensation [2,3,4]. Non-linear compensation techniques have also been proposed [5] . Though these and other techniques have demonstrated some benefit, they also have their limitations. Examples include restrictions to a single iteration interval, amplification of high frequency noise resulting in a lack of smoothness in displays, required knowledge of pilot performance, and increased sensitivity to system instabilities. Jewell et. al. [4] have demonstrated some success in balancing the benefits and "penalties" using a frequency domain compensation technique. The common denominator among these techniques is that the compensation is directed towards the simulator software and hardware to minimize the effects of temporal delays.

Since there is a human in the loop, a compensation technique that takes advantage of the human capabilities and limitations could be the basis for improving operator control performance when operating with transport delays. Related work with aircraft simulators has given indications that this may indeed be the case.

Miller and Riley added full motion cues to a flight simulation task of following a sinusoidally oscillating target and recorded a significant improvement in task performance. [6] Calspan researchers have demonstrated similar results using an in-flight simulator. They found that the addition of actual flight motion to a fixed-base simulation improved pilot handling quality ratings. [7]

Previous studies have also investigated the use of visual cues to improve control performance. Junker used cues based upon computer-generated abstractions of real-world scenes to improve performance in a roll-axis tracking task. [8] Augmentation of supplementary feedback has also been investigated for an approach to landing task and has demonstrated some success. [9]

Methodology

Subjects

Forty two non-pilot subjects participated in the experiment. These subjects were parsed into seven groups of six with the use of a single axis critical tracking task (CTT). The CTT has proven useful in past studies to obtain homogeneous groups of subjects.

Experimental Design

*Member, AIAA

The experiment used a 3 x 2 between-subject design. The two independent variables were the supplementary cue devices (CUE) and supplementary cue delays (CUEDEL). The variable CUE had three levels; (1) Dynamic seat motion cue, (2) Attitude Indicator cue, and (3) Peripheral horizon cue. The two levels of CUEDEL were transport delays of 67 ms and 200 ms for the supplementary cues. The primary display always had a transport delay of 200 ms. See Figure 1 for a block diagram of the experimental design.

Apparatus

Two type of simulators were used in this study; a fixed-base simulator and an in-cockpit motion simulator. Both simulators were run on a DEC PDP 11/60 digital computer. Both utilized visual displays that were generated by a raster-graphics, high-resolution Silicon Graphics IRIS 3120.

A block diagram of the human-vehicle control system for the fixed based simulator is shown in Figure 2. The motion-base task simply adds the dynamic seat as a second display driven by the equations of motion.

The dynamic seat of the Advanced Low Cost G-Cuing system (ALCOGS) in-cockpit motion device[10] was used for the motion supplementary cue condition (Figure 3). The dynamic seat consists of a movable seat pan and seat back that includes five degrees-of-freedom; pitch, yaw, and roll rotations and x and z translations. The seat pan and back are constructed of rigid plates which are controlled by a hydraulic servomechanism with a bandwidth in excess of 10 Hz. This hydraulic servo has a phase characteristic equivalent to about a 20 ms pure time delay.

In this study only the rotational degrees-of-freedom were used. The seat was programmed so that angular displacement, in both pitch and roll, was 0.333 of the angular displacement (aircraft pitch and roll angles) displayed on the screen. Yaw was a coupling effect controlled by the amount of roll.

Aircraft Dynamics

The dynamics were of a fighter-type aircraft flying at 3,048.8 meters (10,000 ft) at a speed of 128.5 meter per second (250 kts) This aircraft was approximated by a first-order filter with a break frequency of 2.85 rad/sec in the roll axis and by a second-order filter with a break frequency of 6.3 rad/sec in the pitch axis.

Table 1: Aircraft Dynamics

Roll Rate Dynamics

$$\frac{1}{\tau_R s + 1}$$

Pitch Rate Dynamics

$$\frac{(\tau_{\theta 2} s + 1)\,\omega_{sp}^2}{s^2 + 2\zeta_{sp}\omega_{sp} s + \omega_{sp}^2}$$

where:

ω_{sp} - undamped natural freq. - short period = 6.3 rad/sec

ζ_{sp} - damping ratio - short period = 0.7

τ_{02} - first order pitch mode time constant = 0.8 sec

τ_R - first order roll mode time constant = 0.35 sec

Coupling between the two axes was included such that when a turn was initiated, an altitude loss occurred unless pitch corrections were made.

The subject's control inputs were made with an isometric control stick mounted on the right side of the seat. The static control gains for the stick were set for a response of 1.45 deg/sec (derived from 3 lbs. per g) and 14.89 deg/sec for pitch rate and roll rate, respectively, when a one pound force was applied at a point 9.5 cm (3.75 inches) up from the base of the stick (constant airspeed of 250 kts). The stick gains were set to record up to a 10.5 lb. maximum force input in either the pitch or roll axis with a breakout force of 0.5 pounds.

Delay Verification

To measure the transport delay, several test frequencies were substituted for stick command inputs. A photocell was used to measure the differences in display luminance. The phase difference between the input to the aircraft dynamics and the output measured by the photocell was determined by a frequency analyzer (Bafco model 916). The phase lag due to the aircraft dynamics was subtracted from the measured phase difference at each of the test frequencies. The transport delay was then calculated by dividing the adjusted phase difference by the test frequency. The reader is directed toward a more detailed description of this process in Johnson et.al. [11]

Primary Display

The primary visual display contained the information required to perform the task. This display had a transport delay of 200 ms for all subjects. The task was a disturbance regulation task where the subjects were required to maintain an altitude of 30.6 meters (100 ft) over a flat terrain grid and remain parallel with the longitudinal lines. During the trials, the subjects flight path was perturbed by an approximation of a Dryden gust model.

The pilots were seated 0.61 meters (24 inches) from a 0.28 by 0.38 meter (11.5 X 15.5 inch) color monitor. This provided a 26 degree high by 35 degree wide field of view. The monitor was a color, raster-scan SRL monitor with a 1000 line by 1024 pixel resolution.

The scene consisted of a black grid on a green terrain background. The grid extended out 1829.2 meters by a width of 457.3 meters (6000 X 1500 ft). Each individual element of the grid was 122.0 meters by 30.5 meters (400 X 100 ft). As the simulated aircraft flew over the grid, the lateral lines moved toward the subject thus providing linear flow cues.

The display presented the rectangular grid to the pilots in a perspective view above a flat terrain. Altitude cues were provided as the longitudinal lines splayed in or out depending on the direction of the altitude change. The longitudinal grid lines were used for heading maintenance. Examples of this grid are provided in Figure 4.

Supplementary Cues

The supplementary cues used in this experiment are found in the actual flight environment. An effort was made to portray them as accurately as the hardware and software would allow. Past studies have demonstrated that the primary display used in this experiment provides subjects with good intrinsic flight cues. [12] The supplementary cues provided attitude information only and therefore did not present the subject with any altitude or heading cues. Thus the subject could not depend on the supplementary cues to perform the task and neglect the essential intrinsic cues provided on the primary forward display.

As stated before, the three supplementary cues were the dynamic seat motion, an attitude directional indicator (ADI), and peripheral displays. The dynamic seat was explained in a previous section. The ADI was located on a second monitor 15 degrees below the line-of-sight to the initial horizon location on the primary display.. At the distance of 0.61 meters the ADI presented a 12 by 15.5 degree field of view. This is representative of current commercial aircraft cockpit CRTs. The peripheral displays were provided using two 0.28 by 0.38 meter color monitors, one on each side of the subject. These monitors presented a horizon line where a white sky and a black ground met. The aft edge of the peripheral displays and point of pitch rotation were coincident and were located 0.61 meters (24 inches) from the design eye point and 72.5 degrees from the center of the primary display. Figure 5 illustrates this experimental configuration.

Disturbance

A wind gust model was used to perturb the aircraft pitch and roll during a trial. This model was driven by the sum of ten discrete, harmonically-unrelated sinusoids for each axis. This disturbance was an approximation of a Dryden gust model with an RMS gust amplitude of 13.6 ft/sec (moderate gust). The frequencies of the spectra for the pitch and roll axis were interleaved with each frequency separated by at least 1/4 octave from adjacent frequencies (Table 2). These interleaved disturbance frequencies for pitch and roll allow for the investigation of control strategies in each axis separately.

Procedures

The subjects were parsed into seven groups of six subjects; six experimental and one control group. The control group was not provided with a supplementary cue device and used the same primary display with a transport delay of 200 ms. Each subject performed a series of critical tracking tasks (CTT) prior to group placement. Past studies have demonstrated high correlations of CTT scores and asymptotic control performance for this flight task. The CTT scores were used to assign the subjects to homogeneous groups.

Each subject was then given a training session where he or she received instructions on how to perform the task and an explanation about the performance scoring. The subjects then completed 50 trials by performing two, five trial sessions per day over the next five days. After each trial, the subject was given immediate knowledge of results of their performance. From previous research, subjects have reached asymptotic performance for this task upon completion of 50 trials.

The dependent variables collected were root-mean-square (RMS) heading and altitude error scores. Control stick time histories were also collected. In particular, the number of times the subjects went from a positive to a negative command (or visa versa) were summed for both pitch and roll inputs.

Results

The RMS data were logarithmically transformed prior to data analysis. The advantage of this data transformation technique is it reduces the heterogeneity of error variance of the subject groups. A second-order polynomial regression model was used to fit the data since it was observed that approximately one third of the subjects reached asymptotic performance early in their training. Towards the end of their 50 trial training sequence, it was observed that performance of several of these subjects was beginning to deteriorate. This we believe was due to a loss of motivation in the task after the subject had reached asymptotic performance. Efforts are being taken to correct this problem in future studies. From the model results, the point of minimum predicted RMS and the rate of learning (the slope from the initial point to the minimum point) were used for analysis purposes.

An analysis of variance (ANOVA) was first performed on the minimun predicted RMS scores to determine if there were any differences between supplementary cues (CUE), cue delays (CUEDEL), and their interaction. Table 3 contains the summary table for this analysis. It can be seen that there was a significant difference between the 67 and 200 ms delay groups. No differences were indicated between the supplementary cues.

A second ANOVA was performed to determine if there were differences between the experimental conditions and the control condition. Table 4 contains the summary table and Figure 6 illustrates the differences between conditions including the Tukey's 95 percent confidence intervals. No differences were indicated from this analysis.

The third analysis looked at the learning rates of each of the groups. Table 5 contains the summary table. The slope of the linear section of the curve that fit between the initial point and the lowest point on the second-order regression model was used in the analysis. Again no statistical differences were found.

The subject's asymptotic performance in the CTT (the higher the scores the better), which was used to group subjects into homogeneous groups, was correlated with the minimum predicted RMS error performance and demonstrated a significantly high negative correlation.

The final analysis was to look at the correlation of the stick zero crossings to the performance scores. Here again a significantly high negative correlation existed between these variables and the minimum predicted RMS performance scores. Table 6 contains the results of the above correlation analyses.

Conclusions

The results indicate that for altitude control, the conditions with the faster updating secondary cues produced better RMS error performance than the conditions with the secondary cues at the same rate as the delayed primary display. The trend was the same for heading but was not statistically significant. When compared to the control condition (no supplementary cue) none of the secondary cue conditions produced a statistically significant performance improvement for either heading or altitude control. However, in all cases, RMS error performance was better with the faster updating secondary cues than with the control condition.

These results may be looked at from a different perspective which is also of interest to the simulation community. While we hypothesized that the fast secondary cues would compensate for the delay in the primary display, i.e. improve performance, one might also argue from the opposite viewpoint; that the cue mismatch would lead to a degradation in performance. Clearly this was not the case. Thus from a cue mismatch perspective, these results suggest that for this task, or one that is similar, a 133 ms mismatch will have no significant performance decrement.

A follow-on experiment will investigate if there are any transfer-of-training effects using the peripheral cue condition which consistently turned in the best performance scores.

The correlations between the performance scores and the stick zero-crossings continue to demonstrate a statistically significant negative relationship for each axes. As RMS error scores decrease, stick activity increases.

The single axis critical tracking task also demonstrated its usefulness in parsing subjects into homogeneous groups. A follow-on study will look at the effectiveness of a dual-axis critical tracking task in predicting performance on this type of task.

Acknowledgments

The authors would like to thank Grant McMillan for his assistance and recommendations during this research project. Support for this research was provided by the AAMRL/HEF at Wright-Patterson Air Force Base, Ohio, under contract number F33615-85-C-0541.

REFERENCES

1. Gum, D.R. and Albery, W.B. (1977). Time delay problems encountered in integrating the Advanced Simulator for Undergraduate Pilot Training. Journal of Aircraft 14, 327-332.

2. Ricard, G.L. and Harris, W.T. (1980). Lead/lag dynamics to compensate for display delays. Journal of Aircraft 17, 212-217.

3. Crane, D.F. (1983). Compensation for time delays in flight simulator visual-display systems. Proceedings of the AIAA Flight Simulation Technologies Conference, Niagra, Falls, NY.

4. Jewell, W.F., Clement, W.F., and Hogue, J.R. (1987). Frequency response identification of a computer-generated image visual simulator with and without a delay compensation scheme. Proceedings of the AIAA Flight Simulation Technologies Conference (No. 87- 2435-CP), Monterey, CA.

5. Hess, R.A. and Myers, A.A. (1984). A nonlinear filter for compensating for time delays. Proceedings of the 20th Annual Manual Conference on Manual Control; NASA-ARC, Moffett Filed, CA.

6. Miller, G. K. and Riley, R. R. (1976). The effect of visual-motion time delays on pilot performance in a pursuit tracking task. In proceedings of AIAA Visual and Motion Simulation Conference; Dayton , OH.

7. Bailey, R. E. , Knotts, L. H., Horowitz, S. J., and Malone, H. L. (1987). Effect of time delay on manual flight control and flying qualities during in-flight and ground-based simulation. In proceedings of AIAA Flight Simulation Technologies Conference (No. 87-2370-CP); Monterey, CA.

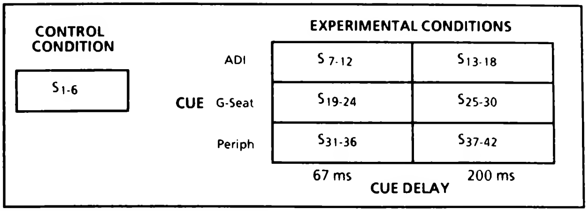
8. Junker, A.M. and Price, D. (1976). Comparison between a peripheral display and motion information on human tracking about the roll axis. Proceeding of the AIAA Visual and Motion Simulation Conference , Dayton, OH.

9. Lintern, G. (1980). Transfer of landing skills after training with supplementary cues. Human Factors, 22 (1), 81-88.

10. Kleinwaks, J. M. (1980). Advanced low cost g-cuing system (ALCOGS). Technical Report No. AFHRL-TR-79-62, AFHRL, Brooks AFB, TX.

11. Johnson, W. V. and Middendorf, M. S. (1988). Simulator transport delay measurement using steady-state techniques. In the Proceedings of the AIAA Flight Simulation Technology Conference. Atlanta, GA.

12. Riccio, G. E., Cress, J. D. and Johnson, W. V. (1987). The effects of simulator delays on the acquisition of flight control skills: control of heading and altitude. In Proceedings Of The Human Factors Society 31st Annual Meeting; pp. 1286 - 1290. New York, NY.



CONTROL CONDITION			EXPERIMENTAL CONDITIONS	
S1-6	CUE	ADI	S7-12	S13-18
		G-Seat	S19-24	S25-30
		Periph	S31-36	S37-42
		CUE DELAY	67 ms	200 ms

Figure 1. Temporal Mismatch Experimental Design

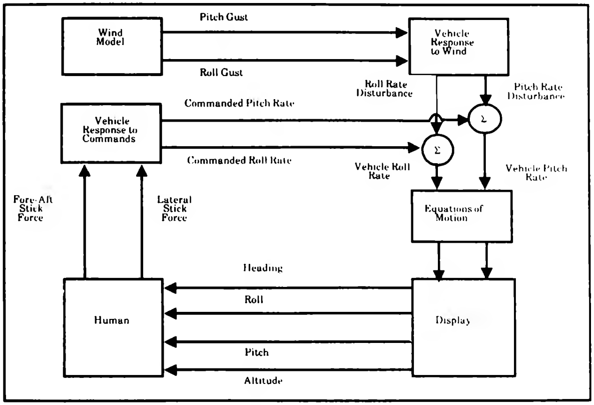


Figure 2. System Block Diagram of Human-Machine System

Table 2. Gust Spectrum Characteristics

Pitch Gust Spectrum			Roll Gust Spectrum		
Frequency (rad/sec)	Correlated Power (dB)	Percent of Total Power.	Frequency (rad/sec)	Correlated Power (dB)	Percent of Total Power
0.276	16.1	3.1	0.460	26.7	18.7
0.644	20.5	8.5	1.012	26.9	19.6
1.565	23.7	17.8	2.117	27.3	21.5
2.669	24.7	22.4	3.405	26.3	17.1
4.326	25.7	28.1	5.430	25.3	13.6
7.271	23.3	16.2	9.296	22.0	6.4
11.781	16.7	3.5	5.002	17.7	2.4
19.420	7.2	0.4	24.666	12.0	0.6
31.017	-2.7	0.04	39.669	5.7	0.2
49.793	-13.7	0.003	62.310	-1.1	0.03

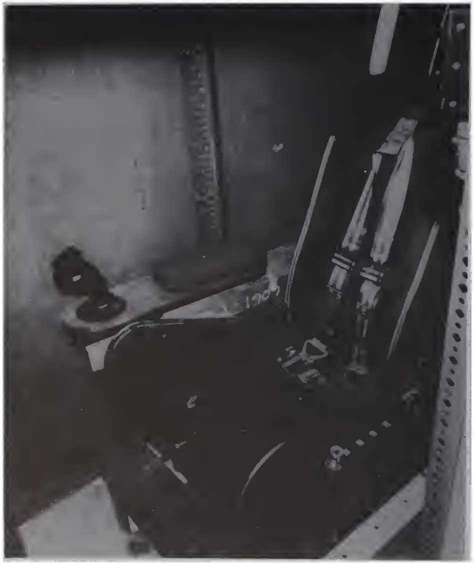
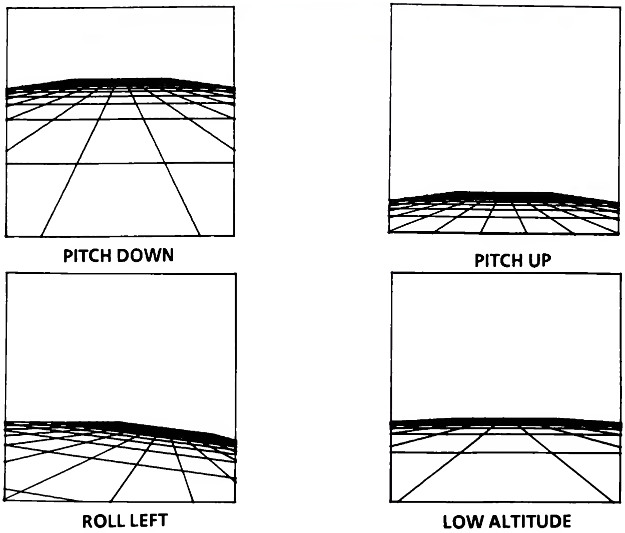


Figure 3. ALCOGS Dynamic Seat



PITCH DOWN

PITCH UP

ROLL LEFT

LOW ALTITUDE

Figure 4: Simple Grid Display

Table 3. Supplementary Cue ANOVA Summary Table

Final Predicted Altitude Score

Source	df	SS	F	p
Cue	2	0.419	0.22	0.8013
Cue Delay	1	0.4955	5.27	0.0289 **
Cue * Cue Delay	2	0.0417	0.22	0.8025
Subject (Cue * Cue Delay)	30	2.8212		

Final Predicted Heading Score

Source	df	SS	F	p
Cue	2	0.1717	0.84	0.4422
Cue Delay	1	0.0115	0.11	0.7394
Cue * Cue Delay	2	0.0341	0.17	0.8475
Subject (Cue * Cue Delay)	30	3.2901		

** statistically significant at $p < 0.05$

Table 4. Comparison ANOVA Summary Table

Final Predicted Heading Score

Source	df	SS	F	p
Condition	6	0.5445	0.85	0.5381
Subject (Condition)	35	3.7215		

Final Predicted Altitude Score

Source	df	SS	F	p
Condition	6	0.5792	1.05	0.4084
Subject (Condition)	35	3.2854		

Table 5. Learning Curve Slope ANOVA Summary Table

Slope of Logarithmic transformed Heading Data,

Source	df	SS	F	p
Condition	6	0.0027	0.64	0.6990
Subject (Condition)	35	0.00028		

Slope of Logarithmic transformed Altitude Data,

Source	df	SS	F	p
Condition	6	0.0023	0.52	0.7874
Subject (Condition)	35	0.00021		

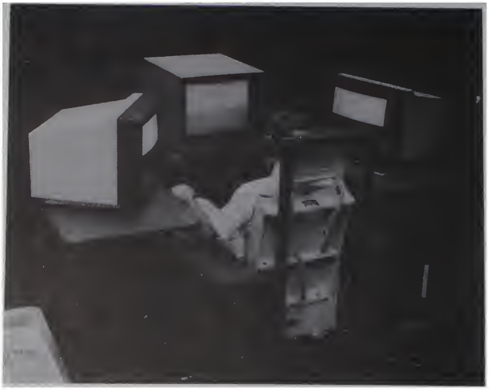


Figure 5. Peripheral Display Configuration

Table 6. Correlation Analyses

Critical Tracking Task vs. Estimated RMS Performance Scores

	Estimated Asymptotic Heading	Estimated Asymptotic Altitude
CTT	- 0.692*	- 0.611*

Control Stick Input Parameters vs. Estimated RMS Performance Scores

	Pitch Zero Crossings	Pitch Rate Reversals	Roll Zero Crossings	Roll Rate Reversals
Estimated Asymptotic Heading	- 0.457*	- 0.264*	- 0.498*	- 0.138*
Estimated Asymptotic Altitude	- 0.559*	- 0.237*	- 0.396*	- 0.102*

* statistically significant at $\underline{p} < 0.0001$

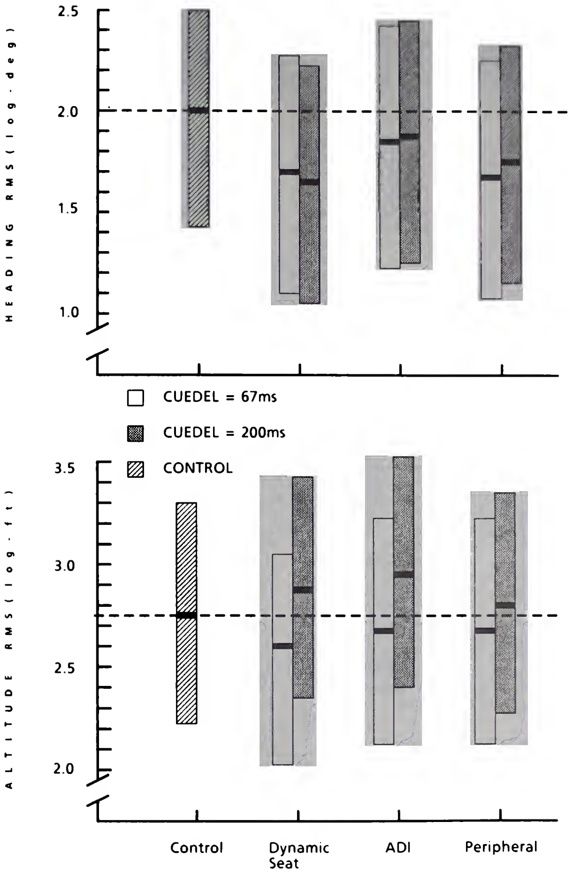


Figure 6. Logarithm of asymptotic performance data with Tukey's 95% confidence intervals.

SPACE OPERATIONS AND SPACE STATION REAL-TIME SIMULATION

D. Hernandez*, A.A. Molineros, W.C. Wagner
Space Transportation Systems Division
Rockwell International
Downey, California

Abstract

This paper describes the Space Operations and Space Station Simulator, a man-in-the-loop, real-time test bed designed to support the analysis, design, and evaluation of future space systems. The state-of-the-art, distributed, hybrid simulator was developed and built by the Space Transportation Systems Division of Rockwell International. A high-fidelity facility with prototype Space Shuttle on-board computers and realistic Shuttle and Space Station controls and displays, the simulator is capable of real-time hardware- and human-in-the-loop simulation using embedded and nonhomogeneous processors, cockpit controls and displays, and out-the-window graphic displays. Presented is an overview of the simulation configuration and capabilities, the hybrid computers and mathematical models, and the hardware and software implementation features.

Introduction

The Rockwell Space Operations and Space Station Simulator has performed numerous simultaneous Space Station and Space Shuttle simulations. These real-time exercises employed dual-manned cockpits with displays and controls, high-fidelity simulation of vehicle dynamics and controls, remote manipulators, out-the-window graphics, and space-rated Shuttle on-board computers. The simulator has been used to support several simulation scenarios tailored for analysis and design studies related to man-machine interfaces, operation of remote manipulators, Space Station construction, Shuttle and Space Station operations, satellite-servicing processes, and other activities associated with space applications. The high real-time computational demands of this simulation were met by use of a mix of analog computers, minicomputers and super-minicomputers, array processors, graphic and embedded processors, and computer-based hardware emulators, all under the control of a distributed real-time executive with various computer operating systems.

Space Operations Simulator

The Space Operations and Space Station Simulator was built in two steps. The first built was the Space Operations Simulator, whose development was begun in 1984. It was used initially as an engineering development tool to study the deployment, retrieval, and assembly of the Space Station by Space Shuttle in real time with test pilots in the loop. Incorporating a prototype Shuttle data-processing system with flight software and an instrumented Shuttle aft flight deck, the simulator was used to develop procedures for the early deployment and construction of the Space Station, and to study the man-machine interface. After a Rediffusion POLY 2000e color raster graphics system was installed, the facility was also used as a part-task remote manipulator system trainer employing more realistic and more complex Space Shuttle and Space Station remote manipulator graphics.

The Space Operations Simulator configuration is shown in Fig. 1. This layout includes all the major hardware elements of the simulator: the Shuttle onboard general-purpose computers (GPC's) and their memory interface assembly (MIA) interfaces, manipulator control interface unit emulator, manipulator arm kinematics and dynamics simulator, graphic scene generators and displays, and Shuttle aft crew station controls and displays. A functional block diagram of the simulator is shown in Fig. 2, which delineates the top-level functional requirements to be implemented and allocated to the different computational elements.

Hybrid Computers

A combination of digital and analog computers was used to implement the simulator (Fig. 3). Two Data General (DG) Eclipse S-250 host processors with three Eclipse S-120 satellites and one array processor are used to simulate the dynamics equations, functions, and interfaces of the remote arms. The DG processors are interfaced through the use of multicomputer communication adapters (MCA's). The host programs run under Data General's real-time disk operating system, and the satellite processor programs run under the real-time operating system. An Electronics Associates PACER computing system, consisting of a PACER 100 digital processor and a PACER 781 parallel analog processor, is interfaced with the Eclipse processors and with the aft cockpit through analog-to-digital and digital-to-analog converters. Continuous parallel processing, the inherent operating structure of the analog computer, gives the PACER 781 a unique combination of real-time computing power and accuracy in solving the servo models of the remote manipulator system (RMS) joints.

Data-Processing Subsystem

The Space Shuttle's on-board data-processing subsystem provides processing capabilities for guidance, navigation, and control; communications and tracking; displays and controls; system-performance monitoring; payload management and handling; remote manipulator control; and other selected subsystem functions. Accepting input commands and data from the Shuttle crew, the on-board sensors, and external sources, the subsystem performs computations and generates output commands and data to accomplish the on-board data processing.

The data-processing subsystem functions of interest in the Space Operations Simulator are those pertaining to the payload and the on-orbit functions of the remote manipulators, which are controlled and operated from the Shuttle aft cockpit. Ref. 1 describes a Space Shuttle hardware evaluator that used a forward cockpit to evaluate the other subsystem functions during orbital, entry, and landing operations. The Shuttle data-processing subsystem elements used in the simulator during on-orbit operations are shown in Fig. 4.

*Member AIAA

Published by American Institute of Aeronautics and Astronautics, Inc., with permission.

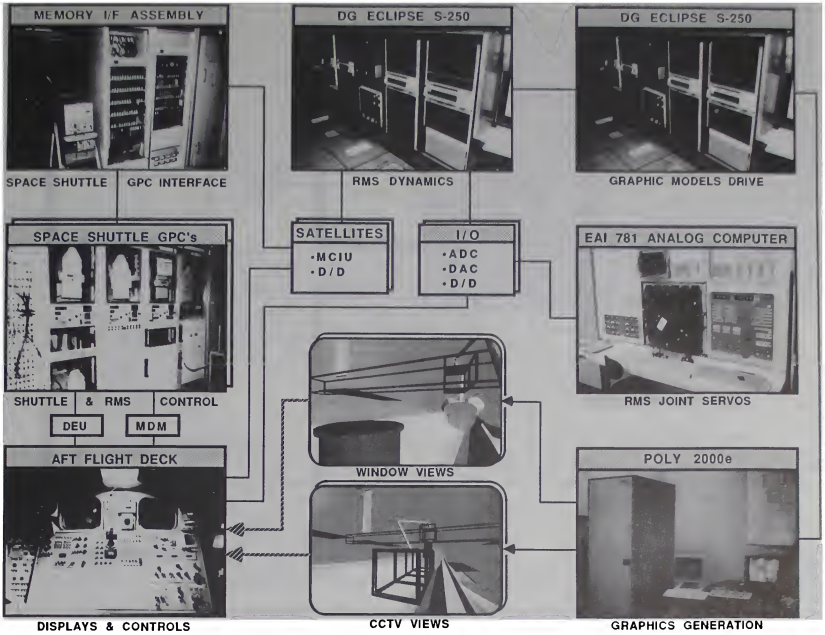


Fig. 1 Space Operations Simulator configuration.

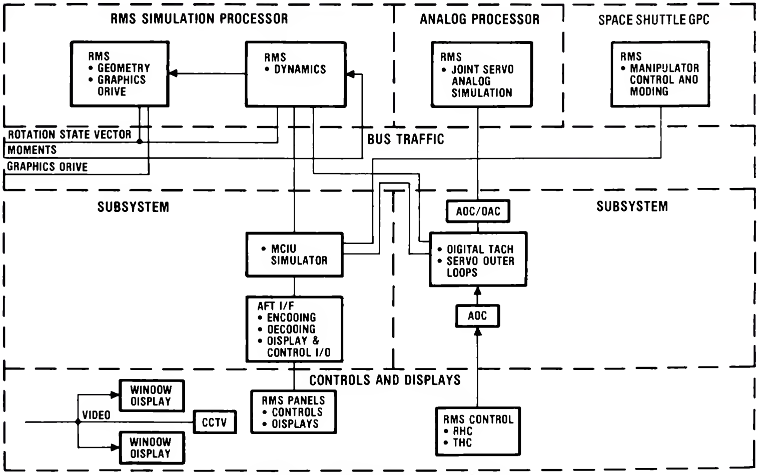


Fig. 2 Space Operations Simulator block diagram.

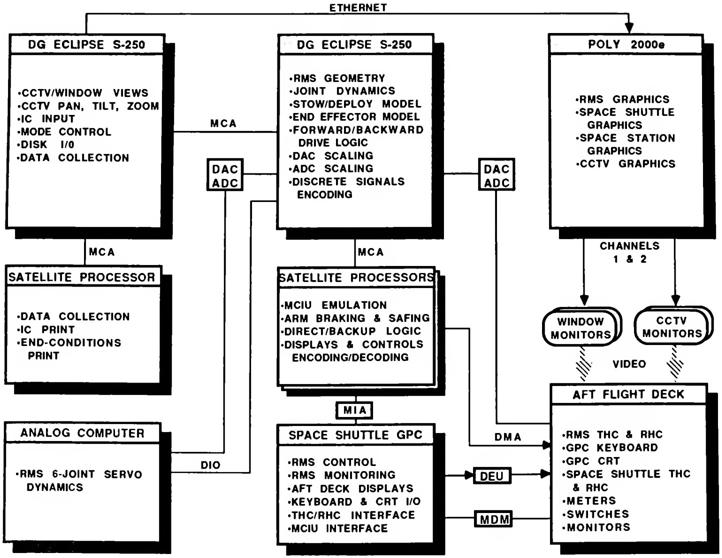


Fig. 3 Space Operations Simulator math model allocation diagram.

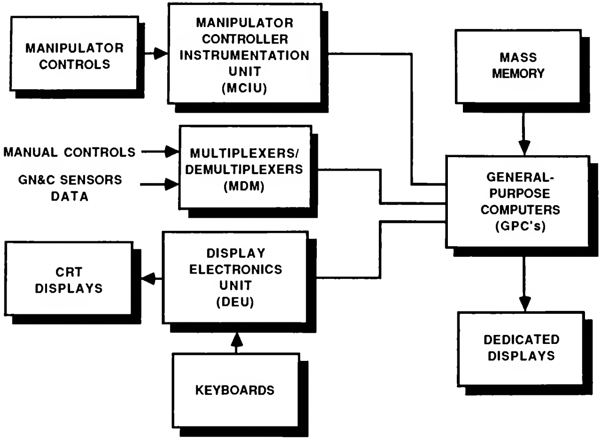

MANIPULATOR CONTROLS
MANIPULATOR CONTROLLER INSTRUMENTATION UNIT (MCIU)
MASS MEMORY
MANUAL CONTROLS
GN&C SENSORS DATA
MULTIPLEXERS/ DEMULTIPLEXERS (MDM)
GENERAL-PURPOSE COMPUTERS (GPC's)
CRT DISPLAYS
DISPLAY ELECTRONICS UNIT (DEU)
DEDICATED DISPLAYS
KEYBOARDS

Fig. 4 Space Shuttle data-processing subsystem block diagram.

Special-Purpose Hardware and Interfaces

A special-purpose manipulator control interface unit (MCIU) emulator was designed and built to perform functions similar to those of the Space Shuttle MCIU, such as interfacing with the aft cockpit RMS controls and the Shuttle on-board computer, but in this case it communicates to a hybrid computer simulation of the RMS. This is a complex emulator with high-frequency computational requirements, since it is part of the control loop of the six RMS joint motors. The aft cockpit controls and displays are front-ended with a special-purpose microprocessor to perform data conversions and buffering and to interface with the digital simulation processors.

Aft Flight Deck Controls and Displays

The aft flight deck controls and displays are shown in Fig. 5. The deck consists of payload, mission, and on-orbit stations, but only the latter is active in the simulator. The on-orbit station contains the rendezvous, docking, payload, and RMS controls. A separable forward flight deck was built to support parallel and independent simulation activities. However, the forward deck is not part of the simulator, and the Shuttle pilot-in-the-loop control is performed from the on-orbit station. Display and control instrumentation was designed to emulate the real hardware.

Aft Flight Deck Visual Display System

The aft flight desk visual display system provides the RMS operator with the out-the-window visual scenes and closed-circuit television (CCTV) displays required to perform RMS operations. The simulator's aft flight deck is shown in Fig. 6. The deck's out-the-window views are of the RMS and the orbiter cargo bay, and of the Space Station when it is in view. The simulated CCTV provides views of five television cameras located in the RMS wrist and elbow and in the cargo bay. Two Megatek MG5000 graphics display systems were used initially, but they were replaced by the Rediffusion POLY 2000e dual-channel system.

Out-the-window visual displays show scenes to the RMS operator through the orbiter overhead and aft windows. The correct display and field of view in each window are simulated, and 25-inch TV monitors display the visual scenes generated through a POLY 2000e channel in the overhead and aft windows.

Fig. 6 Inside view of aft flight deck on-orbit station.

The second channel of the POLY 2000e provides the orbiter RMS CCTV displays. There are five simulated TV cameras available for space operations, two of which can be selected for viewing by the RMS operator using the orbiter video-switching controls. All cameras have simulated zoom, pan, and tilt capabilities. The RMS arm and elbow cameras move with the arm, while the other three are fixed in the cargo bay, their locations depending on mission needs. Two 8-inch monochrome TV monitors display the selected simulated camera images to the RMS operator.

POLY 2000e Graphics System

This system presents realistic 3-D, color, smooth-shaded video images of objects, space vehicles, and robotic arms that are all changing under the control of equations coded in the system and in the simulation host computers, updating them 30 times a second to provide an illusion of smooth motion. By performing most computations for 3-D graphic displays in hardware, microcode, and the resident processors, the POLY 2000e frees the host computers of these tasks, minimizing the hosts' need to monitor and update the graphics processing.

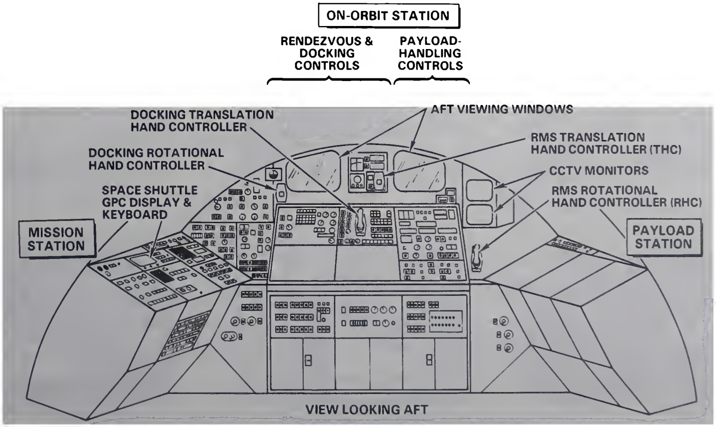

ON-ORBIT STATION
RENDEZVOUS & DOCKING CONTROLS
PAYLOAD-HANDLING CONTROLS
DOCKING TRANSLATION HAND CONTROLLER
AFT VIEWING WINDOWS
DOCKING ROTATIONAL HAND CONTROLLER
RMS TRANSLATION HAND CONTROLLER (THC)
CCTV MONITORS
SPACE SHUTTLE GPC DISPLAY & KEYBOARD
RMS ROTATIONAL HAND CONTROLLER (RHC)
MISSION STATION
PAYLOAD STATION
VIEW LOOKING AFT

Fig. 5 Space Shuttle aft flight deck controls and displays.

The fundamental function of the POLY 2000e hardware is to display polygons in real time and in three dimensions, after performing all required polygon transformations and display operations in special hardware and microcoded high-speed bit-slice processors that have been optimized for high throughput. Up to 2,000 polygon primitives can be displayed in real time from a much larger data base of polygons. Fig. 7 depicts four representative graphic displays generated by the POLY 2000e for simultaneous Space Station and Space Shuttle simulation and operations.

RMS Math Model

The RMS arm assembly, shown in Fig. 8, consists of seven rotational joints: shoulder swing-out, shoulder yaw, shoulder pitch, elbow pitch, wrist yaw, wrist roll, and wrist pitch. When deployed, the swing-out joint is held fixed during operations. The other joints contain a servo-controlled motor that must be sent rate and angle commands.

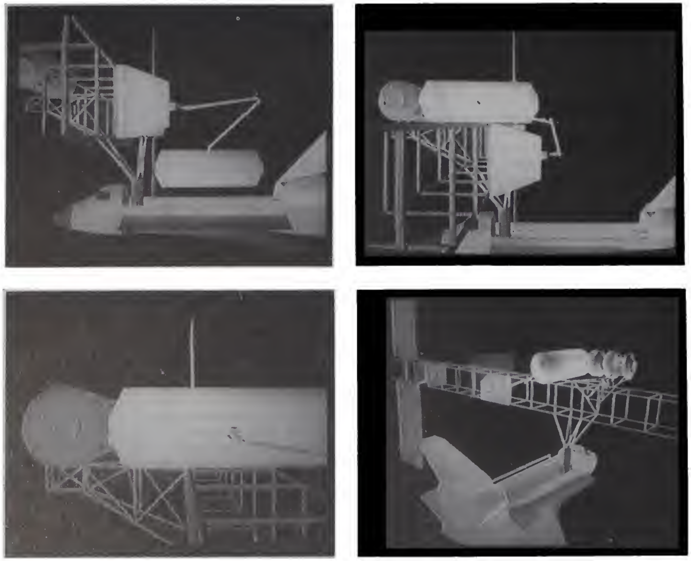
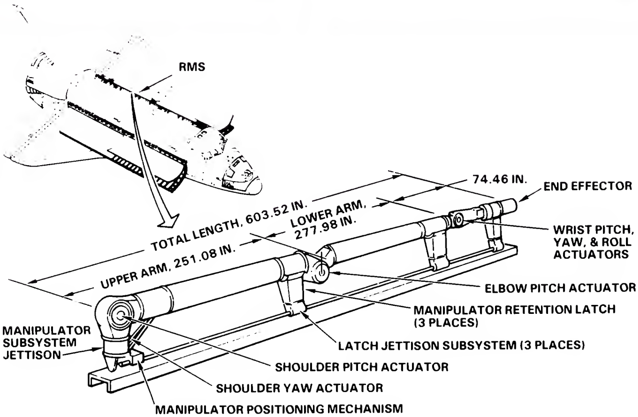


Fig. 7 Views of POLY 2000e-generated graphics.



RMS
74.46 IN.
END EFFECTOR
TOTAL LENGTH, 603.52 IN.
LOWER ARM, 277.98 IN.
UPPER ARM, 251.08 IN.
WRIST PITCH, YAW, & ROLL ACTUATORS
ELBOW PITCH ACTUATOR
MANIPULATOR RETENTION LATCH (3 PLACES)
LATCH JETTISON SUBSYSTEM (3 PLACES)
MANIPULATOR SUBSYSTEM JETTISON
SHOULDER PITCH ACTUATOR
SHOULDER YAW ACTUATOR
MANIPULATOR POSITIONING MECHANISM

Fig. 8 Remote manipulator system arm assembly.

Research was performed to arrive at a set of equations for the RMS dynamic simulation. Fletcher, Rongved, and Yu[2] applied a vectorial mechanics approach for the study of a two-body satellite connected at a hinge. Hooker and Margulies[3] generalized the results of Ref. 2 to develop the equations of motion for a cluster of n-interconnected rigid bodies arranged in the form of a "topological tree." Hooker[4] succeeded in explicitly eliminating constraint torques associated with physical constraints existing at the connections between bodies, thus arriving at the minimal set of equations. Likins[5] presented a systematic method for obtaining the equations of motion in a scalar form based on the developments in Refs. 3 and 4. The RMS has a "topological chain" structure during the payload-handling phase. In this simulation, the method presented by Likins for a cluster of (n + 1) bodies in the form of a tree is used to arrive at the equations of motion for (n + 1) bodies in a chain. The proof of the results in Ref. 5 can be found in Refs. 4 and 6. The set of equations arrived at and the steps involved therein can be applied to the RMS by properly specifying the number of bodies, the joint axis, location of the joints, and the masses and moments of inertia of the Shuttle, the payload, and the arm segments, etc. The method of applying these equations to simulating the dynamics of the RMS is described in Ref. 7.

Simulated in addition to the RMS dynamics were the arm geometry, the end effector, the joint servos and sensors, and the arm cameras. The models are described in Ref. 7.

Math Model Allocations

High-fidelity models of the RMS joint servos were developed and implemented using hybrid computers and special-purpose emulation hardware. Shown in Fig. 3 are the allocation of RMS models and their interfaces with the on board processor, the special-purpose subsystem emulators, the cockpit controls and displays, and the graphics processors. The models are solved in real time at sufficient computational speeds to prevent a significant degradation of the model solutions, the on-board Shuttle GPC computations, or the pilot-in-the-loop performance.

Space Station Simulator

To satisfy the requirements of combined Space Shuttle and Space Station operations, the Space Operations Simulator was expanded by the addition of a Gould 9781 dual-processor computer, a second Rediffusion POLY 2000e graphics system, two IBM PC AT's, a Space Station cupola, a portable work station simulator, and a node flight deck with associated displays and controls (Fig. 9). Multiple vehicle dynamics with control systems and sensors are flown in a simulated environment to provide the interactive space objects and the multivehicle relative geometries that drive the computer-generated out-the-window views. These views are simultaneously displayed in the Shuttle aft cockpit displays and in the 180-degree-field-of-view cupola windows. A layout of the Space Opera-

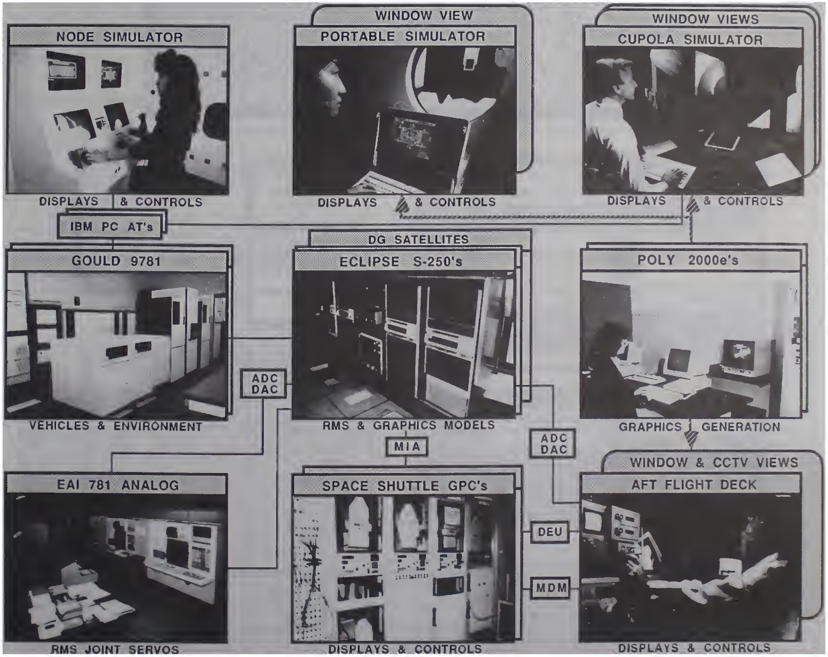


Fig. 9 Space Operations and Space Station Simulator layout.

tions and Space Station Simulator math models is shown in Fig. 10. A touch screen man-machine interface provides a flexible, reconfigurable switch-panel control. The simulator employs modular software techniques with generic subsystem models to provide, quickly and efficiently, an environment in which to evaluate candidate controls and displays.

The combined simulator configuration in Fig. 11 depicts the major hardware elements that comprise the closed-loop system. A Gould 9781 dual-processor computer was added to host all the pertinent algorithms that generate the dynamic scenes simulating the interrelationship between the Space Station and Space Shuttle, the natural environment, and other orbiting spacecraft. The Data Gen-

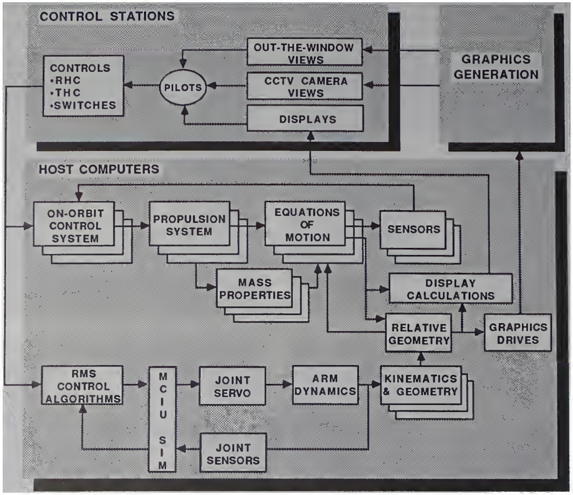


Fig. 10 Space Operations and Space Station Simulator math models.

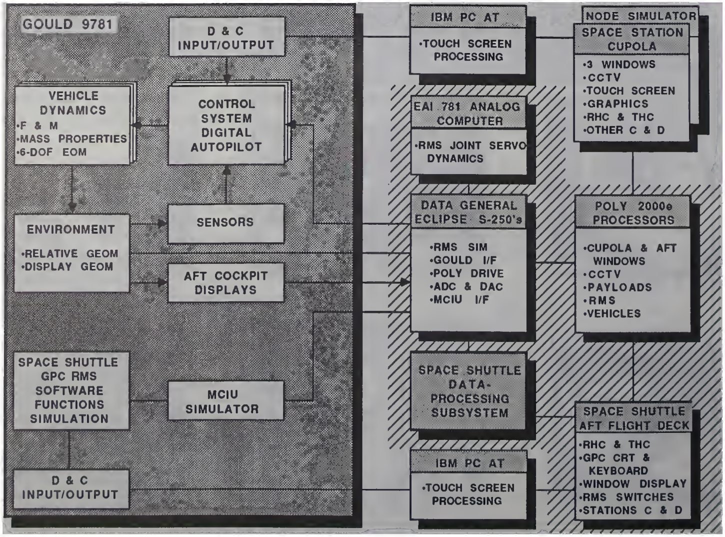

GOULD 9781
D & C INPUT/OUTPUT
VEHICLE DYNAMICS •F & M •MASS PROPERTIES •6-DOF EOM
CONTROL SYSTEM DIGITAL AUTOPILOT
ENVIRONMENT •RELATIVE GEOM •DISPLAY GEOM
SENSORS
AFT COCKPIT DISPLAYS
SPACE SHUTTLE GPC RMS SOFTWARE FUNCTIONS SIMULATION
MCIU SIMULATOR
D & C INPUT/OUTPUT
IBM PC AT •TOUCH SCREEN PROCESSING
EAI 781 ANALOG COMPUTER •RMS JOINT SERVO DYNAMICS
DATA GENERAL ECLIPSE S-250's •RMS SIM •GOULD I/F •POLY DRIVE •ADC & DAC •MCIU I/F
SPACE SHUTTLE DATA-PROCESSING SUBSYSTEM
IBM PC AT •TOUCH SCREEN PROCESSING
NODE SIMULATOR
SPACE STATION CUPOLA •3 WINDOWS •CCTV •TOUCH SCREEN •GRAPHICS •RHC & THC •OTHER C & D
POLY 2000e PROCESSORS •CUPOLA & AFT WINDOWS •CCTV •PAYLOADS •RMS •VEHICLES
SPACE SHUTTLE AFT FLIGHT DECK •RHC & THC •GPC CRT & KEYBOARD •WINDOW DISPLAY •RMS SWITCHES •STATIONS C & D

Fig. 11 Combined-simulator configuration block diagram.

eral Eclipse S-250's now also serve as interface front-end computers, transmitting stream data between the Gould 9781 and the Rediffusion POLY 2000e processors and transmitting data to and receiving data from the analog computers via digital-to-analog and analog-to-digital converters.

The real-time simulation stations—the Space Station cupola and the node and portable work station simulators—serve as man-machine interfaces for the Space Station simulation. These stations contain all the controls and displays required to perform real-time Space Station operations. The displays and controls include CRT monitors, touch screens that provide a multiwindow capability, and rotational and translational hand controllers needed to simulate 6-degree-of-freedom control in yaw, pitch, and roll rotations and in x, y, and z translations.

Cupola and Node Simulators

The Space Station cupola was designed to serve as a control outpost, like a traffic control tower at an airport. It has two primary functions: to aid and direct the space operations of the various vehicles flying near and around the Space Station (Fig. 12) and to teleoperate some of these vehicles and the station's remote manipulator systems. In order to satisfy these requirements, the cupola simulator had to be designed and built to properly simulate the displays and controls in real time, to conduct the necessary simulation scenarios, and to present realistic real-time visual representations of all the space objects associated with Space Station activities.

The Space Station node simulator (Fig. 13) is a windowless work station that has all the capabilities of the cupola except out-the-window visual cues. Instead, four monitors permit the simultaneous viewing of up to four TV camera and graphic scenes.

Controls and Displays

The cupola translational and rotational hand controllers, which provide Space Station and spacecraft attitude maneuvering, are also used to control the remote manipulator system in all its computer-augmented and manual operational modes.

A touch screen panel was added to the cupola displays to optimize switch configurations, data displays, and mode-control capabilities of all the bodies associated with different missions. The touch panel displays the default setup configuration, which the pilot modifies to select the various system capabilities. For example, by operating the rotational hand controller, the pilot could make the three window-projection systems that display a 180-degre field of view rotate the image for a full wraparound view of the computer-simulated space objects. The objects could consist of a partial star field background, the Space Station, the Shuttle, and other spacecraft, such as the orbital maneuvering vehicle, manned maneuvering unit, and a free-flyer. The operator could also select, through the touch screen panel, any of these spacecraft in order to perform translational and rotational maneuvers via remote control, while at the same time the touch panel would display the associated data for the vehicle selected, such as position, velocity, and relative rates. The touch panel also gives the operator the freedom to select various points of view within the computer-generated scene, where simulated TV cameras are located. As the camera view is selected, the instantaneous view of the selected area of the graphics data base is displayed on a monitor in front of the operator. Since this view incorporates the effects of camera zoom, pan, and tilt, it is very realistic.

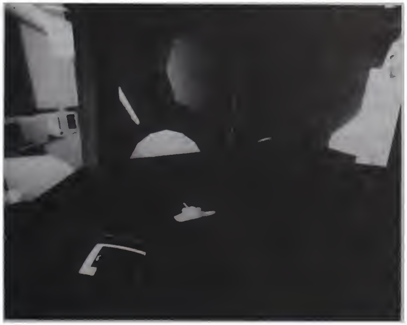


Fig. 12 Out-the-window view from Space Station cupola.

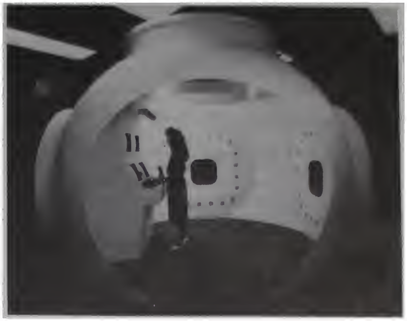


Fig. 13 Space Station node simulator.

Another CRT monitor is dedicated to the GPC keyboard panel and mode display capability, and a mouse-driven pointer is used to select the various GPC functions that control the RMS movement in a computer-augmented mode.

Control Station Input/Output

Each Space Station control station has a translational and rotational hand controller along with simulated switches to change subsystem modes and states. The cupola and node simulators use a multipage touch screen display to model switch functions. The hand controller and switch buffers are routed to the appropriate function via special input/output routines for each control station. A priority hierarchy scheme is used to arbitrate asset contentions.

The touch screen pages are modeled in IBM PC AT's connected to the Gould 9781 through serial RS-232 communication lines. The switch page models are totally generic in their design. Input files are used to configure each page for both graphics and functions, with each switch position mapped to the specific output variable and value. Reconfiguration of a page requires only a change to input tables. There are switch pages for cupola/mode control, control and moding of each RMS, control mode and gain selection for each vehicle control system, and selection and control of simulated TV cameras.

GPC Math Model

The Shuttle GPC's, which are scheduled laboratory assets in high demand, are not always available. To maximize productivity and flexibility, steps were taken to augment the Shuttle GPC's with a functional math model equivalent. The principal Shuttle GPC functions emulated are the RMS control laws, which are hosted in the

Gould 9781. Input/output to this emulated GPC is through the same MCIU buffers, allowing the RMS servos and aft flight deck displays to be driven by either the real or the emulated GPC. The emulated RMS control laws were modified to control serially either the port or starboard Shuttle RMS arm or the "fixed" or "moving" Space Station RMS teleoperators.

Vehicle Dynamics and Control Math Models

The goal of the Space Station Simulator expansion was twofold: to restructure the Space Operations Simulator around space vehicles and payloads that exhibited the proper orbital mechanics and to provide a means to control those bodies using reaction control systems and/or servo-driven remote manipulator arms. The initial objective was to provide an early capability to evaluate man-machine interaction with typical vehicles, and later to expand the fidelity of certain models for unique vehicles and subsystems.

A generic module approach was taken for the rapid development of an initial capability, along with a simulator architectural approach employing arrays for vehicle state vectors and subsystem input/output variables. The generic vehicle subsystems consist of a "performance-configured" on-orbit control system that fires 12 generic reaction control jets to produce disturbance forces and moments. The system configures the thrust levels and locations of the jets to produce the appropriate responses in each vehicle when the pilot specifies rotational and translational acceleration performance values along with control system rate limits, dead bands, and gains for each vehicle. The generic on-orbit control system supports acceleration, pulse, and discrete rate/attitude hold mode for each rotation and translation axis, as well as dynamic gain and limit selection. The vehicle sensors are models of the inertial measurement unit and rate gyros.

The array architecture of the simulation models allows the vehicle and payload state vector variables to be indexed, thus permitting relative positions, velocities, and attitudes to be easily computed. When a vehicle or payload is captured by a manipulator arm or grapple device, its state vector is computed relative to its host vehicle. Upon release of the object, the equations of motion are dynamically reinitialized to the vehicle state at the instant of release.

The vehicles modeled were a Space Shuttle orbiter, the Space Station, an orbital maneuvering vehicle, a manned maneuvering unit, and a telerobot. Each vehicle had unique mass properties and performance characteristics. After the initial capability was achieved, fidelity of the Space Shuttle on-orbit control subsystem was increased by modeling the Shuttle-unique requirements of 44 reaction control jets and a nonlinear control-law phase plane. This was accomplished within the structure of the 12-jet generic system, thus limiting the modifications to only the control system module.

Results

High-fidelity models and displays were implemented in nonhomogeneous distributed processors, complemented with special-purpose hardware emulators and interfaces, using prototype hardware in the loop. Models and software were allocated to specialized processors (e.g., analog, vector, scalar, graphics, embedded) to take advantage of the computational capabilities that they offer in solving certain types of algorithms. The synchronization, execution, and communication problems among the processors were solved by using a combination of operating systems, a distributed executive, and high-speed communication channels. The correct environment and interfaces, with the appropriate timing and frequency response, were provided for an embedded flight computer that was part of the closed-loop system interacting with pilots in the loop. The Space Shuttle, the Space Station, the orbital maneuvering vehicle, and other spacecraft were simulated by models with sufficient capabilities and fidelity to perform realistic space operations, such as rendezvous, proximity maneuvers, approach, and docking. The simulated Space Station and Space Shuttle remote manipulator arms were used in the design and evaluation of procedures for assembling the Space Station, handling payloads, capturing and berthing co-orbiting spacecraft, and assembling large space structures.

Conclusion

An effective and productive simulation with high-fidelity models, realistic cockpit displays and controls, a complex Shuttle data-processing subsystem, elaborated interfaces, and out-the-window displays was developed and built within cost and on schedule, and operated in real time with test pilots at the controls. The high-fidelity modeling requirements and the intensive real-time computational demands of the simulator were met through the use of a concurrent, parallel, distributed-processor configuration with a limited set of nonhomogeneous computer resources, effective model allocations and computer utilization, specially designed hardware emulators, and high-speed interfaces. This simulator was successfully used to design, evaluate, and refine external space operations—rendezvous, proximity maneuvering, docking, capturing, berthing, and assembly—involving multiple spacecraft with complex dynamics, controls, and remote manipulator systems.

References

1. Hernandez, D. *Space Shuttle Hardware Evaluator Man-in-the-Loop System.* AIAA Flight Simulation Technologies Conference, Long Beach, California, June 1981.

2. Fletcher, H.J., L. Rongved, and E.Y. Yu. "Dynamics Analysis of a Two-Body Gravitationally Oriented Satellite," *Bell System Technical Journal,* Vol. 42, No. 5 (1963), pp. 2239-2266.

3. Hooker, W.W., and G. Margulies. "The Dynamical Attitude Equations for an n-Body Satellite," *The Journal of the Astronautical Sciences,* Vol. XII, No. 4 (1965), pp. 123-128.

4. Hooker, W.W. "A Set of r Dynamical Attitude Equations for an Arbitrary n-Body Satellite Having r Rotational Degrees of Freedom," *AIAA Journal,* Vol. 8, No. 7 (July 1970), pp. 1205-1207.

5. Likins, P.W. "Hybrid-Coordinate Spacecraft Dynamics Using Large-Deformation Modal Coordinates," *Astronautica Acta,* Vol. 18, No.5 (1973), pp. 331-348.

6. Likins, P.W., and G.E. Fleischer. *Large Deformation Modal Coordinates for Non-Rigid Vehicle Dynamics.* Jet Propulsion Laboratory, Technical Report 321565 (1972).

7. Hernandez, D., A.A. Molineros, and E. Ramos. "IR&D G.O. 84517 Final Report." Rockwell International Corporation, Internal Letter PDL-84-SSSD256 (October 4, 1984).

A Design Methodology for Neutral Buoyancy Simulation of Space Operations

88-4628-CP

David L. Akin*
Massachusetts Institute of Technology
Cambridge, MA

Abstract

Neutral buoyancy has often been used in the past for EVA development activities,but little has been done to provide an analytical understanding of the environment and its correlation with space. This paper covers a set of related research topics at the MIT Space Systems Laboratory, dealing with the modeling of the space and underwater environments, validation of the models through testing in neutral buoyancy, parabolic flight, and space flight experiments, and applications of the models to gain a better design methodology for creating meaningful neutral buoyancy simulations. Examples covered include simulation validation criteria for human body dynamics, and for applied torques in a beam rotation task, which is the pacing crew operation for EVA structural assembly. Extensions of the dynamics models are presented for powered vehicles in the underwater environment, and examples given from the MIT Space Telerobotics Research Program, including the Beam Assembly Teleoperator and the Multimode Proximity Operations Device. Future expansions of the modeling theory are also presented, leading to remote vehicles which behave in neutral buoyancy exactly as the modeled system would in space.

Introduction

Underwater simulation has traditionally been used in the space program for specific elements of extravehicular activity (EVA) planning, such as hardware design evaluation or procedures development. In these applications, the purpose of the simulation is to obtain qualitative assessments of hardware designs, or relative performance times for two or more competing procedures. Simulation design practices have focused on "high fidelity"; that is, making the underwater hardware conform *physically* as exactly as possible to the flight hardware. No widespread use of neutral buoyancy has yet developed for space simulations beyond EVA applications. This paper details the research experience of the Space Systems Laboratory in neutral buoyancy simulation of space operations, focusing on the areas of EVA assembly of large space structures, and robotic operations including telerobotic structural assembly and vehicle flight control.

The choice of neutral buoyancy for space simulations must be made relative to the task considered and the merits of the other potential simulation media. For space operations, the critical features of the space environment to be modeled include the microgravity dynamics and the physical worksite of the task under study. There are essentially four types of earth-based simulation for space operations: parabolic flight, computer scene generation, motion-based facilities, and neutral buoyancy. Each of these has its own advantages and disadvantages, and each is described here briefly to explain the rationale behind the choice of neutral buoyancy for space operations simulation.

Parabolic flight consists of performing a repeated series of climbs and dives in transport category aircraft, most notably the NASA KC-135 aircraft operated out of Ellington Air Force Base by the Johnson Space Center. This system provides the only true weightlessness available on earth, with typical G levels of .01-.03g limited to periods of 20-25 seconds, followed by 90 - 120 seconds of pullout between parabolas at 1.8 - 2g. This short period of weightlessness, along with the physical limitations of the KC-135 cabin and the significant operating cost of the aircraft, significantly restricts the applicability of this simulation method. It continues to be used for a limited set of EVA tasks, primarily IVA procedures training on critical tasks such as pressure-suit donning and doffing.

Computer scene generation can provide excellent results at reasonable costs for tasks in which the human is providing limited-bandwidth control, such as flight control of a remote vehicle. It lacks fidelity in cases of constrained motion, particularly in the modeling of contact forces between vehicles or components. It also restricts the simulated task to those which would typically be performed using a video monitor or through a window, as opposed to "hands-on" operational tasks.

Motion carriage facilities can be subdivided into two types: the traditional motion carriage, where all degrees of freedom of the moving system are actively controlled by computer-controlled actuators, and the flat-floor facilities, where the horizontal degrees of freedom are passively simulated through air-bearing vehicles on specially-constructed test beds. To obtain the remaining three degrees of freedom, flat floor facilities must still provide computer-controlled motion actuators, as in the case of the full motion carriage system. In either case, the actively-articulated degrees of freedom are subject to the limitations of the controller algorithms. In addition, the complexity of the worksite modelled (in terms of number of components and relative positions) is severely limited by the essentially two-dimensional nature of the mechanisms involved in the motion simulation. While some of these facilities have been used for man-in-the-loop control simulations of spacecraft, they share with the computer scene generation systems the fact that they are best suited for independent vehicle control task simulations.

Neutral buoyancy simulation uses the underwater environment to simulate the weightless environment of space. For accurate simulation, each component must be adjusted so that its mass is exactly offset by the mass of water it displaces. For accurate motion, it must further be adjusted such that the center of mass is coincident with the center of buoyancy, so that it has no preferred orientation in the water. Neutral buoyancy has no effective limitations on the number, size, or relative configuration of components. Humans can interact directly with the hardware for extended periods of time, limited only by the physiological limitations of diving at ambient pressures.

The primary drawback of neutral buoyancy simulation is that the dynamic properties do not directly simulate those of space, due to the damping and inertial effects of the water. The design of simulation hardware is also complicated by the corrosive underwater environment, even the relatively benign environment of the large neutral buoyancy tanks of NASA at the Marshall Space Flight Center and the Johnson Space Center. At this time, the "common wisdom" seems to be that neutral buoyancy simulation is almost exclusively useful for extravehicular activity tasks, with all other operational tasks (e.g., IVA and robotics) performed with limited motion-based simulation, or with no allowance for modelling the space worksite environment at all.

In this paper, it will be demonstrated that the relative dynamics of the space and neutral buoyancy environments can be treated rigorously, as well as empirically. The analytical understanding of motion in the two environments yields a greater range of potential applications of neutral buoyancy simulations, including robotics and intravehicular operations.

Extravehicular Assembly of Large Space Structures

In the course of systems analyses of several potential large space projects, including space manufacturing facilities and solar power satellites, the productivity of EVA structural assembly occurred repeatedly as a critical factor for the economic viability of

*Assistant Professor of Aeronautics and Astronautics
Associate Director, MIT Space Systems Laboratory
Member, AIAA

each program. At the same time, the MIT Space Systems Laboratory (SSL) was interested in pursuing experimental efforts in space operations, through the use of ground-based simulations. These parallel trends culminated in the current extensive research efforts of the Space Operations Research Group, within the SSL.

Structural assembly is almost ideally suited for an analytical correlation of neutral buoyancy with the same activity in space. The primary activities are gross motions involving major muscle groups (manipulating masses and moments of inertia), instead of motions requiring fine manual dexterity. To the greatest extent possible, simulations of EVA structural assembly can focus on the physics of the task, rather than on the fine details of manipulation. It is also physically stressful for the test subject, and can therefore be of use in determining the limits of human capability in space. Finally, it is a critical technology for space development, from the space station to solar power satellites and beyond.

Prior to developing a structural assembly simulation, the Space Systems Laboratory engaged in the creation of a mathematical model of human motion in both the space and underwater environments.[1] This model was validated experimentally in the neutral buoyancy environment, and in parabolic flight on board the KC-135.[2] The existence of a model proven accurate in the two environments allowed the correlation of body motions between the two, yielding a set of "validation criteria" to be used in the design of simulation tasks.

This initial model was limited to bilaterally symmetric motion of the arms. Although dynamic effects of the whole body (including pressure suit, if present) were included, articulations were only provided in the arms to model the primary range of body motion in typical extravehicular operations. Of principal interest was the dynamic response of the torso to the arm motion, represented by a "tilt-back" of the body in response to the forward motion of the arms. The angle of this tiltback is labelled α (see Figure 1). Even this limited model produced 240 terms in the differential equation prior to simplification, and resulted in the summary equation of body reaction, which is shown in Figure 2. [2].

Verification of this model is demonstrated in Figure 3. The upper graph shows the measured motion of the arms (with respect to the torso) as a function of time. Since continuous derivatives of this motion were required for the numerical integration of the tiltback equation, the solid line shows the fifth-order polynomial curve fit used to create an analytical model of the measured arm motion. The predicted body reaction motion is shown in the lower graph, along with the measured body motion from the experiment. This sample was repeated over different task speeds, sizes of masses moved, test subject body sizes, and simulation media (both underwater and in parabolic flight) to arrive at the complete body dynamics model, which incorporates the dynamic response equation with models of body inertias and drag forces in the underwater environment.

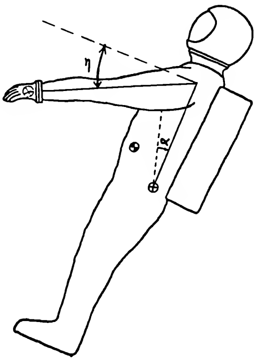


Figure 1
Body Dynamics Model

With the validation of the model in both environments, analytical studies were undertaken to correlate body reaction dynamics to the same motion in each medium. Figure 4 summarizes the most significant results. As may be seen, the fidelity of the resultant body motion in neutral buoyancy is dependent on the mass being manipulated. As long as the test mass (e.g., structural component) is a significant fraction of the test subject's mass, the inertia forces predominate over the viscous, and similar motion occurs in the two environments. This result argues that neutral buoyancy hardware should ideally be of significant mass (as much as 50% of the test subject's body mass), in order to have reasonable (within 20%) dynamic fidelity.

A set of neutral buoyancy hardware was designed and developed in conjunction with the results of the correlation study. The tetrahedral truss configuration was chosen as the simplest self-rigidizing truss configuration possible. Including trapped water, the effective mass of the structural struts were 29.5 kg, as large as possible while retaining sizes and shapes that had a realistic external configuration. The structures were attached to the Space Shuttle cargo bay mockup at a single point, and were designed with sufficient strength to allow effectively unlimited crew motion on the structure during assembly. In line with the focus of this paper on simulation design aspects, the interested reader is referred to other articles [3, 4, 5] for the specific details of the research results.

From the standpoint of simulation dynamics, these assembly tests showed graphically that the critical task was not moving uniform masses linearly, but rotating the 3.6m linear truss elements that make up the structure. For a uniform, cylindrical strut of unit density (required for neutral buoyancy), length L and diameter D, it can be shown that the rotational torque is related to the rotational motion ω by

$$\tau = \frac{\rho}{12} D^2 L^3 \dot{\omega} + \frac{\rho}{8} c_D D L^4 \ddot{\omega}$$

where ρ is the water density and c_D is the section drag coefficient of the strut. The first term of the polynomial is the usual form of the inertial properties; the second term is set by the effect of water velocity distribution along the cylinder. In space, only the first term is present; underwater, both terms contribute to the required torque during motion. As can be seen from this equation, the classical insistence on physical fidelity (making the underwater hardware the same physical dimensions as the flight hardware) results in increasingly disparate torque requirements

$$\ddot{\alpha} = \frac{\dot{\alpha}\left[4Mm\dot{\theta}\,(ld \sin\theta \sin\phi + l^2 \cos\theta \sin\phi)\right] - 2Mm\, ld \cos\phi\,(\cos\theta\, \dot{\theta}^2 + \sin\theta\, \ddot{\theta})}{I_y\,(M+2m) + 2Mm\,(2ld \cos\theta \sin\phi + l^2 \cos^2\theta + d^2)}$$

Figure 2
Body Dynamics Tiltback Equation

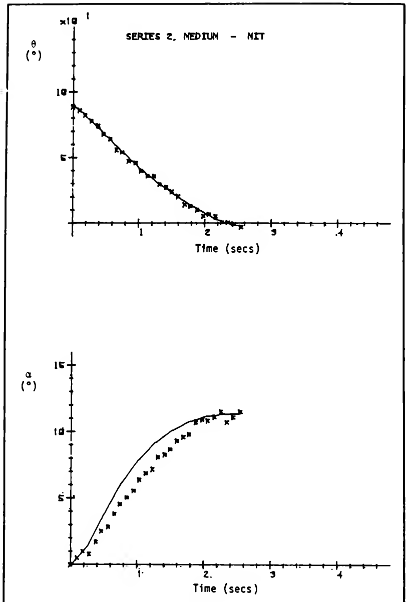


Figure 3
Body Dynamics Experimental Verification

as the size of the object increases.

As the culmination of this activity, the MIT Space Systems Laboratory prepared and flew a structural assembly experiment on space shuttle mission 61-B in December, 1985. This mission consisted of the EVA construction of a tetrahedral truss. Use of 1-cm thick aluminum for the beam wall provided a set of flight hardware which exactly replicated the mass and moment of inertia of the neutral buoyancy hardware. Reference [6] details the results of this flight experiment.

Early simulation dynamics analyses were concerned with the effect of the environment on motion of the crew. Data obtained from the flight experiment allowed a higher-level analysis of crew-imposed loads on the hardware, which further provides more insight into the similarities and differences in the two environments.

Four generic categories of tasks were analyzed: body translation and rotation, and equipment translation and rotation. Stereo imagery from neutral buoyancy training and from flight allowed the reconstruction of three-dimensional trajectories. Numerical estimation of accelerations from these time histories provides an estimate of the time-variance of crew-applied loads. Figure 5 shows the applied load history for equipment rotation in the two environments.

The task considered in Figure 5 was to rotate a structural component through a given angle. However, it is clear from this analysis that the time-histories of applied torque are considerably different. The test subject in space basically applied an impulsive force to start the rotation, then adopted a coasting strategy until the component neared the target orientation. Stepwise torque inputs in the negative direction then reduced the rotational velocity in increments to provide greater control near the desired final orientation. In comparison, the neutral buoyancy beam rotation shows significantly higher initial torques, with velocity decreasing over the "coast" phase due to the effects of water drag. Repeated positive applications of torque kept the beam rotating toward the desired final orientation. No torque was applied to stop the beam, as the water drag quickly reduced the rotational velocity to zero.

As described in Reference [7], complete results from this analy-

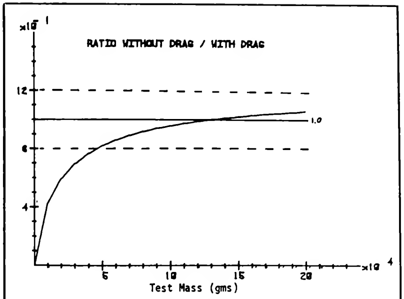


Figure 4
Body Dynamics Validation Criterion

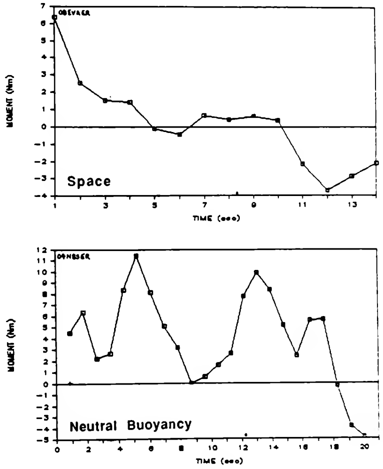


Figure 5
Beam Rotation Torques in Neutral Buoyancy and Space

sis indicate the validity of the concept of dynamically scaling structures for neutral buoyancy simulation of space operations. Figure 6 shows the correlation of maximum applied moment with time to complete the rotation of a Space Station structural strut, both in space and underwater. As this figure shows, a nominal rotation in space (requiring 3 N-m of maximum torque and 14 seconds) would require over 90 N-m of peak torque to match the 14 sec rotation time in neutral buoyancy. If the peak torque were held constant, the underwater rotation would require over 60 seconds, thus severely compromising the time estimates gained from neutral buoyancy simulation. The most likely result from using the flight configuration hardware would be that the crew exerts more force, and also takes longer to complete the rotation. By properly scaling the strut size for neutral buoyancy (by shortening the length to 2.15 m), the dynamically scaled hardware will require the same peak torque to rotate in the same amount of time as the 5 m strut would in space. The pattern of applied torque is still different, with little decelerating torque applied underwater, but nonetheless the overall correlation is much improved over the use of identical hardware for neutral buoyancy and space.

Space Telerobotics Operations

The significance of the preceding section of this paper lies in the rigor of the analysis, rather than the application: EVA simulation has been the predominate (if not total) role of neutral buoyancy simulation for years. In this section, these results will be extended to new applications of neutral buoyancy simulation.

With the success of the EVA studies, the Space Systems Laboratory decided to extend its experimental effort into the field of telerobotics. In the study of Space Applications for Automation, Robotics, and Machine Intelligence Systems (ARAMIS), the SSL concluded its Phase 1 studies with the significant conclusion that telerobotics and telepresence technologies held great potential for future space applications. It was decided to develop an experimental capability in these fields, to provide data on operational applications of these technologies, and to allow meaningful quantitative assessments of the potentials of these technologies in space industrialization.

In large part, the rationale behind SSL telerobotics simulation paralleled that of the EVA research. While much good telerobotics research is ongoing in laboratories across the country, the focus of the MIT work was selected as integrated systems testing, rather than component technology development. If this work were to be performed in the laboratory, it would be limited to small-scale manipulative tasks, with limited motion-carriage correction of the task dynamics, or with the alternative of no dynamics correction of all. Rejecting those possibilities, the same four simulation media described earlier were considered for telerobot systems testing. Because of the desire for maximum work-site validity, along with the existence of a large number of high-fidelity mockups left over from EVA training and the ready accessibility of water tanks (including the MIT Alumni Swimming Pool), the decision was made to commit to the development of space telerobotic vehicles designed for the neutral buoyancy environment.

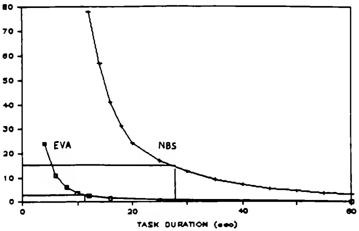


Figure 6
Torque–Time Relations for Space Station Strut Rotation

One advantage of complex, "intelligent" vehicles is that they have the capability for self-controlled motion, which in turned presents the potential for active control of the vehicle's motion to compensate for the unrealistic dynamic effects of the underwater environment. For any body in motion underwater, the predominant forces are those externally applied (for example, vehicle thrust) and those due to hydrodynamic forces, such as drag. The acceleration of the vehicle is due to the sum of these forces, or, writing acceleration as the first derivative of velocity,

$$m\dot{v} = T - \frac{1}{2}\rho v^2 A c_D$$

This nonlinear first-order differential equation may be solved to find an equation for v as a function of time and of the initial conditions

$$v = \frac{v_o\sqrt{a_o\beta} + a_o \tanh\left[\sqrt{a_o\beta}\,(t - t_o)\right]}{\sqrt{a_o\beta} + \beta\, v_o \tanh\left[\sqrt{a_o\beta}\,(t - t_o)\right]}$$

where the constants a_o and β are defined as

$$a_o = \frac{T}{m}$$

$$\beta = \frac{\rho A c_D}{2m}$$

If rest initial conditions are assumed (v_o=0 at t_o=0), this equation simplifies to

$$v = \sqrt{\frac{a_o}{\beta}}\tanh\left(\sqrt{a_o\beta}\; t\right)$$

This equation holds for vehicles in the neutral buoyancy environment. By comparison, a vehicle in space operates in a conservative field, with no damping at all; although this would lead to unlimited velocities, flight rules generally specify limits on velocities in the vicinity of the shuttle orbiter or the space station. Figure 7 shows a typical example of vehicle motion in space and underwater. The sample case chosen is for an existing vehicle: the Manned Maneuvering Unit, here assumed to have a mass of 250kg and a thrust of 50N. Assuming the mass and thrust remain constant in the neutral buoyancy simulation, Figure 7 shows that the underwater version cannot replicate the maneuvering performance of the flight version, even with space velocity limits. The water drag terms slow the rate of acceleration as velocity increases, because an increasing proportion of the thrust must go to counteract water drag. This situation is aggravated if the neutral buoyancy system ignores the presence of virtual mass terms, which are due to the acceleration of water in the proximity of the underwater device. The effect of this is an increase in the effective inertial mass of the vehicle beyond its displacement, further decreasing the acceleration rate. It is clear from this anal-

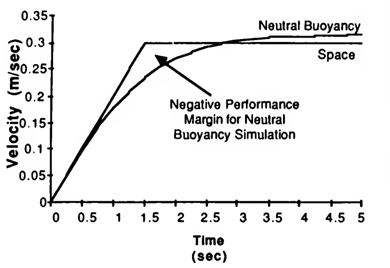


Figure 7
Manned Maneuvering Unit Performance

ysis that excess thrust capability is required for an accurate neutral buoyancy simulation of a space vehicle. The velocity profile for the underwater vehicle must lie outside the space vehicle curve throughout the time history, so that excess authority in the propulsion system may be used to make the underwater simulation fly as the actual vehicle would in space.

Vehicle Mobility Research: Routine operations around a space station will require a substantial amount of vehicle mobility. For example, satellites to be returned to the station for refurbishment or repair must be tracked, approached, captured, returned to the station, and berthed. It would obviously be impractical to use the space shuttle orbiter for such a task; indeed, use of the orbiter would preclude revisiting the vast majority of satellites. For this reason, vehicles such as the Orbital Maneuvering Vehicle (OMV) and Orbital Transfer Vehicle (OTV) have been proposed for early development, and indeed form one of the essential pieces of space station infrastructure.

To maximize the productivity of the limited crew of the Space Station, these orbital vehicles are designed to be flown unmanned. It has been proposed to use teleoperator control of these vehicles in the terminal stages of rendezvous and docking. Teleoperation can result in a variety of problems, due primarily to inaccuracies in communications, such as limited band width and time delays for signal propagation. These effects exist to varying degrees whether the operator is onboard the space station or on the ground. In addition, human factors aspects such as controllability, mental workload, and control station design for the weightless environment are other necessary research areas. An alternative approach to vehicle control is to fully automate all vehicle operations, including the docking phase. In order to further study the effectiveness of the various concepts of vehicle control, the MIT Space Systems Laboratory has developed the Multimode Proximity Operations Device (MPOD).

The multimode portion of the name refers to operating control modes of the vehicle. It is apparent that the lack of sufficient bandwidth and quality of sensory cues to the operator of a remote vehicle will result in the degradation, or total loss, of control capability. In order to examine the widest possible range of sensory information to the operator, MPOD was designed to be an optionally manned vehicle. Shown in Figure 8, the MPOD contains a cockpit for a single operator. Onboard video systems provide a forward view, transmitted to a video monitor on the control panel. In the manned mode, the operator sits internally, wearing common scuba gear with an on-board air supply. In teleoperator mode, the control panel and control interfaces are removed from the vehicle, and the operator in scuba gear can control the vehicle externally, but underwater, using the same control panel and controls as in the onboard case. This eliminates the effects of operator interface differences on comparative runs.

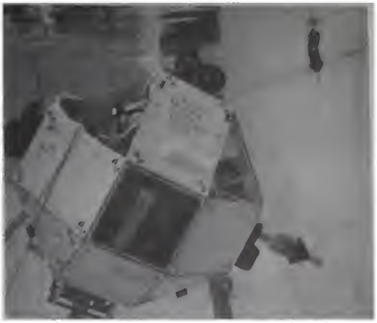


Figure 8
Multimode Proximity Operations Device

The external configuration of MPOD is designed specifically for the maneuvering task of interest. As a symmetrical, pseudo-spherical vehicle, the device exhibits uniform drag characteristics in any direction. With ducted propellers mounted on each of the six principal faces, all six degrees of freedom may be controlled with a set of six thrusters. Two such sets are used on MPOD, both to provide redundancy and to insure that sufficient excess thrust is available for modeling the mobility characteristics of any particular space vehicle. Greater details of MPOD are given in Reference [8].

In order to verify the mass and drag properties of MPOD, the vehicle was towed with a lightweight steel cable under a constant tension. Vehicle dynamic response was obtained by taking data from an optical encoder system which measured line travel. By examining the resultant data, the MPOD effective mass (including virtual mass) and drag coefficient could be directly calculated. The results indicated an inertial mass of 2919kg (as compared to a calculated displacement of 1888kg), and a drag coefficient of 0.674 referenced to a cross-sectional area of 1.705m^2. Figure 9 shows the MPOD performance limits, referenced to a generic Orbital Maneuvering Vehicle velocity profile. It is apparent from this figure that MPOD has sufficient thrust authority to simulate this OMV, given the proper instrumentation and control system to calculate and negate the effects of water drag.

Telerobotic Structural Assembly: In order to obtain data on telerobotic assembly of the same structures used in the prior EVA experiments, the MIT Space Systems Laboratory designed and constructed the Beam Assembly Teleoperator, or BAT. The Beam Assembly Teleoperator, shown in Figure 10, is designed primarily for dexterous manipulation. In its current configuration, it consists of a dexterous 5-degree of freedom manipulator arm and two specialized grappling arms, mounted on a mobility unit with six unlimited degrees of freedom. Operator feedback consists of views from two stereo camera pairs, as well as normal manipulator and vehicle feedback parameters. Primary application of BAT has been in the area of large space structure assembly. BAT has successfully assembled the EASE structure, in both master-slave teleoperation and with some low-level supervisory control capabilities. It has assembled four different structural connectors, including those used in the EASE and ACCESS space structures. It has also been used to assemble Space Station-type truss structures, as well as to investigate the poten-

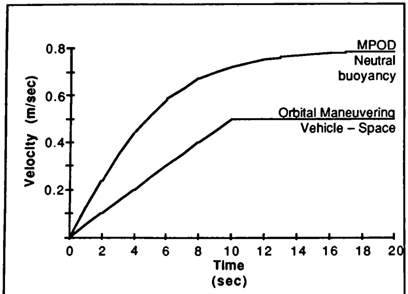


Figure 9
MPOD Performance Characteristics

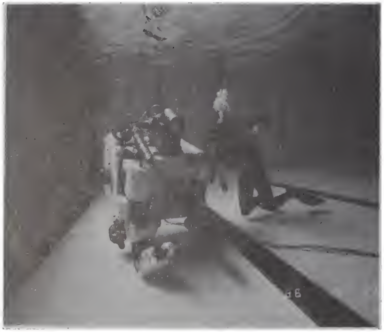


Figure 10
Beam Assembly Teleoperator

tials for human-telerobot cooperative activities in the EVA worksite. Indeed, one of the greatest advantages of neutral buoyancy simulation is in the potential to have meaningful interactions with EVA test subjects. Details of BAT operational test results may be found in Reference [9].

The more advanced capabilities of BAT call for more sophistication in the techniques of neutral buoyancy simulation. For example, one area which will be of interest in the future will be free-flying robotic vehicles in the grappling task, or (potentially) performing manipulative activities on a neighboring vehicle while free-flying in stationkeeping mode. By use of an instrumented wrist reading contact forces between the telerobot and the target vehicle, realistic vehicle dynamics may be simulated and driven through the use of vehicle thrusters. Similarly, a vehicle state vector estimate, if it could be obtained in the underwater environment, could be used to remove the water drag effects from the vehicle response to control inputs. Again, the net effect would be to have a vehicle which behaves (in the point of view of the human or computer operator) as if it were in space, wherein the water drag effects are negated in the closed-loop control system of the vehicle, driven by an internal model of desired vehicle dynamics. Research into the necessary sensor technology and control system algorithms to implement these capabilities is currently on-going in the Space Systems Laboratory.

Conclusions

Neutral buoyancy, as a simulation medium, is capable of far more important applications than its traditional role in the qualitative evaluation of EVA.As has been shown, once an understanding of the underlying fundamental nature of both neutral buoyancy and the space environment are known, much can be done in the design of the simulation hardware and protocol to maximize the correlation between the two media. Experiments can be designed specifically to the phenomena of interest, such as human body dynamics or applied forces and torques.

Neutral buoyancy capabilities are greatly expanded when considered for use with advanced technology systems, such as telerobotics. The use of powered thrusters to control the vehicle's motion underwater opens intriguing possibilities for high-fidelity motion in the environment, closely replicating the actual motion in space for even quite complex activities. While problems remain in analysis and hardware implementation, the benefits of neutral buoyancy as a simulation media argue for a greatly expanded role for this simulation technique in the future. It is not exaggeration to assert that the neutral buoyancy tank will be to space exploration what the wind tunnel was to aviation.

Acknowledgements

The author would like to acknowledge the pivotal roles of Dr. Mary L. Bowden and Dr. Daniel Cousins in the research described in this paper. The primary support for this research has been grant NAGW-21 from the Office of Aeronautics and Space Technology, NASA Headquarters.

References

1 Marian S. Tomusiak, "Simulation of Zero-G Dynamics of Assembly of Large Space Structures Components" S.M. Thesis, Massachusetts Institute of Technology, June 1978

2 Mary L. Bowden, "Dynamics of Manual Assembly of Large Space Structures in Weightlessness" S.M. Thesis, Massachusetts Institute of Technology, January 1981

3 M. L. Bowden and D. L. Akin, "Underwater Simulation of Human Dynamics and Productivities in Extra-Vehicular Activity" International Astronautical Federation 79-109, Oct. 1979

4 D. L. Akin and M. L. Bowden, "EVA Capabilities for the Assembly of Large Space Structures" International Astronautical Federation 82-393, Oct. 1982

5 D. L. Akin, "Quantifying Human Performance in Space Operations" International Astronautical Federation 86-024, Oct. 1986

6 David L. Akin, et.al., *Final Report of the Experimental Assembly of Structures in EVA* Space Systems Laboratory Report, in preparation

7 Daniel Cousins, "Biomechanics of Extravehicular Activity and its Neutral Buoyancy Simulation" Ph.D. Thesis, Massachusetts Institute of Technology, June 1987

8 David L. Akin, "Telerobotics: Neutral Buoyancy Simulation of Space Applications" Undersea Teleoperators and Intelligent Autonomous Vehicles, MIT Sea Grant College Program, January 1987

9 David L. Akin, "Telerobotic Assembly of Space Structures: Results from Neutral Buoyancy Simulation" *Journal of Aerospace Engineering*, in publication

TELESCIENCE TESTBED PILOT PROJECT: EVALUATION ENVIRONMENT FOR SPACE STATION OPERATIONS

88-4629-CP

Michael J. Wiskerchen*
Stanford University
Stanford, CA 94305

Barry M. Leiner**
Research Institute for Advanced Computer Science
Ames Research Center
Moffett Field, CA 94035

Abstract

The NASA's Office of Space Science and Applications (OSSA) has initiated a pilot program to validate the user-oriented rapid-prototyping testbed approach which addresses a range of operations and information system issues. Fifteen universities, under subcontract to the Universities Space Research Association (USRA) and in cooperation with various NASA Centers, are conducting a variety of scientific experiments emulative of the scientific research of the space station era and aimed at resolving critical issues involving space station operations concepts and information system design. The goal is to allow scientists and engineers to interact with potential space station technologies in a manner that will allow resolution of design and specification questions without having to wait until space station hardware is available.

This paper describes the structure and methodology of the rapid prototyping efforts and reports the results for the first eleven months of the 15 university telescience testbed program. In addition, the multi-media networking capabilities between the NASA Centers involved in space station design and operations and the universities are discussed in terms of overall requirements for telecommunications between space station testbed/simulation facilities and the telescience testbed effort.

I. Background

The space station era has thrust NASA into an environment where the systems to be designed and developed are dominated by utilization and long term operations requirements rather than R & D engineering goals. The space station is the first space facility with long term operations (> 30 years) objectives where its success will be determined by its productive and efficient use. In addition, the facility is the first long term international, multi-discipline space capability. It will also be built and operated in a period of time where computer and communications technologies, including ubiquitous and high bandwidth packet switched networks, advanced workstations, on-board processing, and automation and robotics, will be evolving rapidly. This calls for a new systems engineering approach for requirements definition and specification along with new concepts in system design which allows for system evolution as utilization objectives and technologies evolve.

To address this issue the Telescience Testbed Pilot Program (TTPP) was initiated. The TTPP is an innovative activity involving fifteen universities, in Cooperation with various NASA Centers, in user-oriented rapid-prototyping testbeds to develop the requirements and technologies appropriate to the information system of the space station era. The purpose of the TTPP is to:

(1) demonstrate the utility of a user-oriented rapid prototyping testbed approach to developing and refining science requirements and validation concepts and approaches for the information systems of the space station era and beyond;

(2) develop an initial set of recommendations for those requirements, concepts, and approaches; and

(3) develop recommendations as to the best approach to conducting such an activity.

The identification of critical issues to be addressed by the Telescience Testbed became the first important task. With constrained financial resources, it was important to select only those problem areas which could be considered the system "long pole" elements in terms of utilization and space science operations. The following were key technical questions which arose in that initial assessment.

- What is the impact of the distribution of users on how the system should be architecturally designed?
- How can access to the variety of required resources be provided in a coordinated manner?
- What is the required user interface to allow scientists to gain access to the resources in a consistent way?
- What is the impact of reduced or intermittent communications?
- What is the interaction of remote control and autonomous operation of an experiment?
- How should the planning and scheduling of multiple activities using common and shared resources be done?
- What are the requirements for authentication, access control, and security, and how can they be best accommodated?

* Sr. Research Associate, Dept. of Electrical Engineering

** Sr. Research Scientist

- What are the required characteristics of the underlying communications networks and what are the supporting networking technologies to be used?

Along with the technical issues, other significant organizational and management questions arose pertaining to the way the Telescience Testbed would interact and interface with the space station program testbed capabilities (e.g. Data Management System (DMS) at Johnson Space Center and the Space Station Payload Testbed Simulator at Goddard Space Flight Center) and the phase C/D work package contractors. Clear distinctions arose between the objectives of the Office of Space Station (OSS) testbeds and the OSSA Telescience Testbed. While the Telescience Testbed followed objectives stated above, the OSS testbeds were carefully constructed mockups where system level and end-to-end testing could be run. The OSS testbeds were being built to be high fidelity functional emulations such that interface testing during system integration can be achieved. In other words, the Telescience Testbed would primarily help to refine and evaluate system requirements and specification questions during the design and development phase while the OSS testbeds primary objectives address end-to-end testing and verification of developed systems.

II. Systems Engineering and the Requirements Game

To successfully achieve the Telescience Testbed objectives, one must understand the importance of the education process which must be fostered between the utilization, the designers/developers, and the technology communities in establishing good system requirements and specifications. All systems engineering methodologies begin with mission requirements definition and specification. Generally, there are three major players in this initial requirements activity, the systems engineer, the system user (either in person or a surrogate), and the technologist. Most space projects use a linear phased approach (mission needs definition, concept exploration, demonstration and validation, full scale development, production, and operations and support) to carry out the system engineering. Although there may be involvement of all three major players in the early phase activities, the system users and technologist have minimal involvement in the later phases. System development project which use this engineering methodology make the basic assumption that system needs and requirements are fully understood and that the technology is identified during the concept exploration phase and that they will remain essentially static during the other phases. Figure 1 illustrates this linear system development cycle.

The process moves efficiently along from system engineering to design to development to test while budget and schedule are managed carefully. System performance is judged against the initial requirements. As the figure indicates, changing user needs or utilization concepts, evolving technology, and operations cost modeling are not allowed to influence the design or development of the system. If the system requirements are not well known in early phases and/or the system technology or operations concepts are dynamically evolving, the operational system will not be functionally satisfactory or cost effective.

Too often the linear approach starts too late to fully define what the system is. Design engineers generally believe that the system is the design and development of the hardware while others may think that the primary objective is the functional operation of the hardware for some purpose. This lack of communication generally results in optimizing the design for the wrong functions. Optimizing for development efficiencies instead of operational efficiencies can many times lead to costly, unproductive, and unusable systems. One common example of this is optimizing for minimum development cost while ignoring the life cycle cost.

The linear approach to system engineering has not been effective in producing technologically aggressive state-of-the-art systems. This is due to the fact that the user is usually insufficiently familiar with the system concept to fully understand the impact of their requirements on the system. In addition the technologist are unable to keep the technology up to date because of the long development life cycles resulting in the fielding of technologically obsolete systems. The basic assumption for iterative systems engineering is that the requirements and technology will be evolving throughout the life of a project. This requires the formulation of a engineering methodology which allows this dynamic

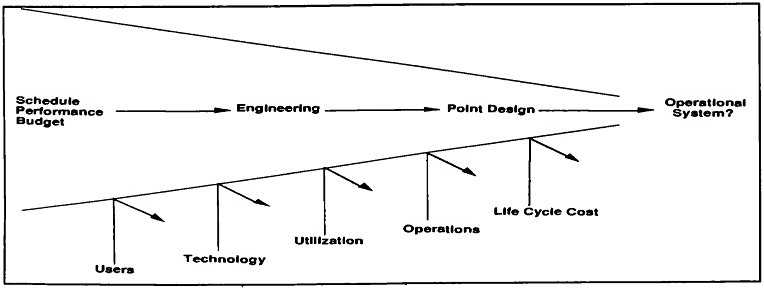


Figure 1. Linear System Development Cycle

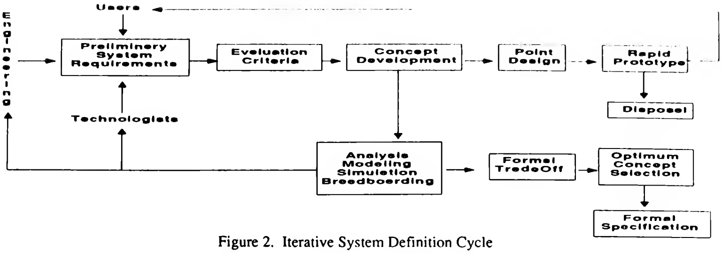


Figure 2. Iterative System Definition Cycle

evolution of requirements and technology to influence the system design and development. Figure 2 indicates such a methodology and that used by the Telescience Testbed Program. The figure includes the engineering process through the demonstration/validation phase. The process begins with the formation of an engineering/user/technologist team to begin preliminarysystem requirements definition from best guess user functional needs. This team should also derive its membership equally from the university, industry and government sectors. Each sector will gain unique benefits from this working level interaction. The team establishes a set of evaluation criteria. These evaluation criteria will be used to select from among the systems concepts that will be developed in the next step. Several different concepts for meeting the previously developed requirements are now developed. The intent is to look at the full range of possible solutions. At this point the concepts can take one of two paths. With either path, the primary objective of the process is to validate the concepts in terms of satisfying the preliminary requirements and to educated the team. Some concepts can be functionally tested analytically in a modeling or computer simulation environment while others must be placed in a rapid prototyping testbed where "quick and dirty" point designs can be operated in a hands-on mode by the team. With both paths, rapid iteration is essential to the success of the methodology. When several competing concepts satisfactorily meet the system requirements, then a formal trade-off process must occur to arrive at the optimum concept. Before formal specification can begin, care must be taken to distill all functional specifications from the concepts such that vendor specific specifications from the point designs are removed. Another important point is to remember that the requirements and specifications that are being developed are for the performance of the system and not the design. This process is repeated at lower levels of detail until the full set of system specifications has been developed at a low enough level so that the requirements can be formally given to the functional designers for implementation.

III. Telescience Testbed - Participants and Projects

The selection of Telescience Testbed university participants was based on a combined expertise in OSSA related space science operations and state-of-the-art telecommunications and information system technology. The science discipline areas of interest were broadly defined to include Earth systems sciences, astronomy and astrophysics, life sciences, and microgravity materials processing. The following institutions are represented in the initial Telescience Testbed team.

The fifteen universities, along with the NASA Centers, have conducted a variety of scientific experiments emulative of the scientific research of the space station era and aimed at resolving critical issues in Space Station Information Systems design. The goal is to allow scientists to interact with potential space station technologies in a manner that will allow resolution of design and specification questions without having to wait until space station hardware is

Ames Research Center
Univ. of Arizona
Univ. of Cal., Berkeley
Univ. of Cal., Santa Barbara
Univ. of Colorado
Cornell University
Goddard Space Flight Center
Jet Propulsion Lab
Johnson Space Center
Kennedy Space Center
Lewis Research Center
Marshall Space Flight Center
University of Maryland
Massachusetts Inst. of Tech.
University of Michigan
Purdue University
Rensselaer Polytechnic Inst.
University of Rhode Island
Smithsonian Astronomical Ob.
Stanford University
University of Wisconsin

Figure 3. Telescience Testbed Participants

available. The following is a short synopsis of some of the testbed experiments currently ongoing as part of the pilot program.

University of California, Santa Barbara is exploring teleanalysis of large dynamic data sets for earth sciences. This investigation includes the test and evaluation of data interchange standards and knowledge based techniques for assisting remote access.

University of Colorado is investigating the cooperative use of data from the Solar Mesosphere Explorer for coordinated measurements, remote access and control. It is also applying a user-workstation-oriented control concept to a number of telescience experiments collaboratively with other program participants.

Purdue University is evaluating teleanalysis concepts using the Purdue Field Spectral Database accessed by a variety of small computers. It is also investigating methods for conducting campaign style experiments and computer data security issues.

University of Michigan is experimenting with teleoperations of a Fabry-Perot Spectrometer combining human with autonomous control, forward simulation techniques to support telerobotics, and the effects of varying time delays in the control loop.

University of Wisconsin is providing a bridge from NSFnet to Mcldas, and using that bridge to merge SME data with data from GOES for atmospheric science. This includes the provision of access to merged data from IBM PCs with Enhanced Graphics Displays, but doing so in a way that protects system and utility software.

Stanford University is experimenting with a model Remote Science Operations Center linked to GSFC, JSC and MSFC using real data from Spacelab 2 to test multimedia Telescience workstations and simulate remote control, monitoring and multi-media conferencing.

MIT is conducting two experiments. The first is a Remote Life Sciences Operation using the KSC sled and JSC simulator with multi-media tests and evaluation of real video needs and implementation options. They also are investigating the remote operation of a telescope at Wallace Observatory using a high bandwidth (T1) link and dissemination of data on campus-wide Project Athena network.

The Space Infrared Telescope Facility (SIRTF) team, consisting of Cornell University, Smithsonian Astrophysics Observatory, CalTech, University of Rochester and University of Arizona, are investigating several issues regarding telescience applied to a Space-based astronomical facility. They are evaluating distributed versus resource-centered models for development (teledesign) and remote access. The ability to interchange analysis software and perform in conference mode for design, operations and analysis will be evaluated. University of Arizona has a special interest in remote control and operations of a ground- based telescope to evaluate feasible degrees of automation, allowable time delays, necessary crew intervention, error control and feasible data compression schemes. Cornell University is investigating trade-offs between on-line local processing and processing at the user's home location as well as investigating the feasibility of establishing standard formats and analysis techniques. Smithsonian Astrophysical Observatory is using remote operation of Mt. Hopkins telescope to evaluate data transmission and dissemination options.

University of California, Berkeley, is evaluating remote control techniques for EUVE over local area and wide-area nets to support a distributed development and operations system.

University of Rhode Island is investigating a novel image compression technique with "zoom" capability to help progress from browsing to detailed analysis of selected areas using modest bandwidths from remote sites.

Rensselaer Polytechnic Institute is investigating operational requirements for microgravity materials processing. This includes testbed activities in the areas of required communications, the feasibility of a remote POCC (Payload Operations Control Center) at RPI, and reliability assessment for teleoperations of microgravity materials science.

RIACS (Research Institute for Advanced Computer Science) is integrating various networking and local computing capabilities into a "telescience workstation", intended to provide the local computing environment for telescience.

These experiments all share the characteristic that they are attempting to apply new technologies and concepts of science operation to ongoing scientific activities. In that process, a better understanding will be gained of the future scientific modes of operation and the systems architectures, concepts, and technologies required to support such operational modes. The early testbed results indicated several serious deficiencies in the present and planned capabilities to carry out Space Station operations. This deficiencies are seen in the telecommunications infrastructure (network capabilities) and the systems engineering methodology for establishing interface and interoperability standards. Although NASA has begun efforts (i.e. the NASA Science Internet, Space Station Technical and Management Information System (TMIS)) to establish functional networks for Space Station operations and management, these efforts are not, as yet, integrated into the mainstream Phase C/D contractor work packages. The TTPP testbed activities, along with with its NASA sponsor, OSSA, have begun to address these deficiencies and have produced significant results.

IV. Computer Networks and Space Station Utilization

The value of computer networking capabilities such as electronic mail, file transfer, and remote access to computers has been well established. Each of the Federal agencies is establishing a computer network to serve its community of researchers. In particular, NSFnet, ESnet (DOE), and NSI (NASA) are all being established based on similar requirements and approaches. The NASA Science Internet (NSI) in particular is being established to ensure that satisfactory basic and enhanced networking service is provided in a cost-effective manner through use of a number of networks (including Space Physics Analysis Network (SPAN) and the NASA Science Network (NSN), a new TCP/IP based network). The NSI program is aimed at cost-effectiveness and ubiquitous connectivity through the use of shared communication resources both internally to NASA (using SPAN and NSN) and with other agencies and through the use of interoperability approaches such as gateways between the various networks. In addition, it is the objective of the NSI Program to support non-mission-critical computer networking for the science programs of the agency; to provide a basic level of computer networking service to NASA sponsored research institutions and affiliated agencies; to use NASA data communications facilities efficiently; to perform research and development in network communications technology and define pilot networks for the NSI; and, to work with private industry, the Internet research community and other government agencies in the evaluation and exploitation of new computer networking technology. The TTPP provides the NSI program with an active working group and environment to evaluate and assess the functionality of network architectures, technologies, and interface standards in terms of operational efficiency and productivity.

The space science community is typically multi-disciplinary and multi-agency. Typical science activities require operation across agency boundaries. For example, exploration of global environmental issues requires cooperation amongst oceanographers, climatologists, atmospheric scientists, and earth scientists. Such activities are funded by several agencies including NOAA, NSF, USGS and NASA. NASA must provide effective, high-performance data communications for its astronomy, earth science, space physics, and planetary science investigators and their data sources and computational facilities. Because of NASA's diverse user computing environment, interoperability among heterogeneous components is an important testbed problem to be prototyped by the TTPP. Networking approaches based on discipline specific or agency specific requirements alone will not provide the wide-spread connectivity and interoperability needed by such multidisciplinary activities, nor will it provide for the effective cost-sharing required if the needs are to be satisfied within feasible resources. The TTPP has been addressing the sharing, interoperabilty, and cross-support requirements through joint testbed investigations involving a number of agencies. These testbeds have had the goal of providing a single "virtual" network to all scientific activities during all phases (including the design, development, operations, and analysis phases) of the Space Station program. This network should allow for transparent interaction between scientists and the resources they require, including access to remote computers, databases, experimental laboratories, and other scientists. Such interaction should only be limited by permission to use the resources rather than limitations in the network connectivity. The TTPP activities also confirmed that present and future network capabilities can neither be defined or implemented effectively without strong and interactive involvement of the user communities. The productivity gains of this interaction are dramatically enhanced by having rapid prototyping testbeds for educating the various participants on new network technologies and operations concepts.

The TTPP activities have shown that the connectivity provided by NSFnet and NSN must be extended upwards in the protocol architecture to provide for connectivity between application processes. NSFnet has addressed this issue by standardizing on the DARPA-developed Internet protocol suite (commonly known as TCP/IP and associated protocols) and is planning for an ultimate migration to the protocol suite being standardized by ISO (as part of their effort on Open Systems Interconnect OSI) when they are available

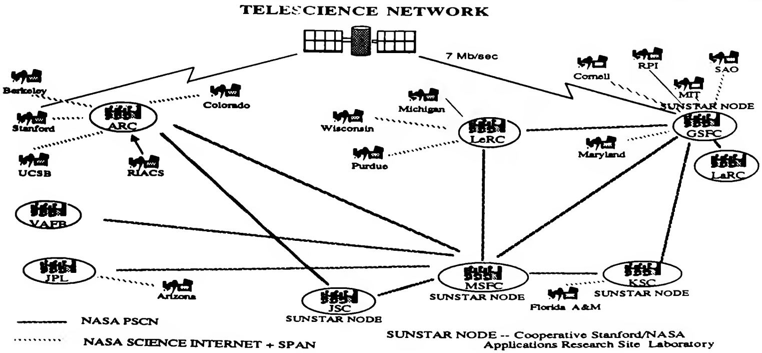


Figure 4. Telescience Network

and are shown to satisfy the requirements of the scientific community. Indeed, it is expected that there will be a widespread migration towards the ISO standard protocols. In the interim, though, the widespread use of the TCP/IP based protocol suite coupled with its use by NSFnet leads to the adoption of these protocols as the standard protocols for the NSN. Furthermore, the similarity between the ISO/OSI and TCP/IP protocols will facilitate the migration to the ISO/OSI protocols.

In the process of providing ubiquitous networking to the scientific community, TTPP efforts have shown that particular attention should be paid to providing the required administrative functions needed for facilitating electronic mail. The TTPP has made heavy use of electronic mail to carry out the distributed program. This started with the development of the initial concept papers on the testbed and continued through today where the activities are coordinated through the use of such structures as monthly informal electronic mail reports. Figure 4 illustrates the present interconnectivity between the TTPP participants.

USRA has attempted to facilitate this ongoing electronic interaction by maintaining a list of electronic mail addresses for the various participants and interested parties, and providing automatic mailing to subsets of interest groups. (For example, a list is maintained for participants involved in earth sciences.) In maintaining this list, USRA has had to validate the various electronic mail addresses to insure that they result in reliable delivery. This has turned out to be a non-trivial task due to the large variety of electronic mailing systems being used (e.g. Internet, SPAN, telemail, nasamail, gsfcmail, OMNET, Bitnet) and the need to deal with changing routing and gateways between systems. For example, the change over from telemail to nasamail caused a considerable effort in assuring accuracy of addresses in the mailing lists.

Based on this experience, we believe that any attempt to provide for and use electronic mail to support multidisciplinary scientific research will require administrative support of the gateways and directory services. Rather than asking the individual scientific researchers or their organizations to provide this function, we believe it would be much more cost-effective to provide such functions on a community wide basis.

A serious deficiency indicated by the initial TTPP studies was the inadequate network connectivity between the science users payload design and development sites and the NASA Center design engineering sites. On-line access to engineering specification and documentation databases is not possible at this time. At the present time, electronic access to engineering information between the various work package contractors is also not possible. The TMIS contract (Boeing-prime contractor) has this as one of its primary task. The TTPP activities indicate that rapid prototypes of the architecture, interface standards, and technologies involved in providing this network capability could be beneficial in producing a productive and efficient solution to this problem. The TTPP participants are working closely with the newly formed industry consortiums (i.e. Open Software Foundation, X-Window Consortium - MIT Athena, CAD Framework Initiative) to have a better understanding of evolving system standards so that the Space Station design will incorporate those standards.

V. Conclusion

The activities of the TTPP university consortium have already dramatically influenced the architectural design and operations concept of the space station. Probably more important, the TTPP has initiated and successfully implemented a new systems engineering methodology for deriving and evaluating system requirements and integrating those requirements into a system design. The TTPP provides an excellent environment, with low programmatic schedule and budget risk, for testing and evaluating new operations concepts and technologies. The close working relationship between the TTPP community and the Space Station Program should insure that the Space Station design provides a utilization environment that is functionally productive and operationally efficient.

THE EFFECT OF PERSPECTIVE DISPLAYS ON ALTITUDE AND STABILITY CONTROL IN SIMULATED ROTARY WING FLIGHT

88-4634-CP

K. A. O'Donnell
U.S. Army Aeroflightdynamics Directorate

W. W. Johnson, C. T. Bennett
NASA Ames Research Center

Abstract

Two simulation experiments investigated the effect of perspective displays on flight performance. In the first, a perspective grid display was superimposed on computer-generated terrain. Subjects attempted to maintain their initial altitude in a simulated hover using terrain and/or one of four grid patterns (e.g. meridian lines, horizontal lines, vertical and horizontal lines, or a quasi-random pattern of dots). Horizontal lines produced the best altitude control performance. The second experiment investigated a square grid in combination with various visual display configurations and grid attachment conditions. The display configurations were: a 120-deg field of view out-the- window display, a 40-deg field of view projected onto a panel-mounted display, and a 40-deg field of view helmet-mounted display. The grid attachment conditions were: no grid, the grid followed movement of the aircraft with respect to pitch, roll, and yaw (complete grid attachment), and the grid remained stationary with respect to pitch, roll and yaw (partial grid attachment). The grid always followed aircraft movement with respect to forward, lateral and vertical translation. Two different visual worlds and two different tasks were employed. Pilots flew through a slalom course and approached to a 40-ft hover from 1500 ft in a world with pylons and a world without pylons. Performance with the panel-mounted display was significantly worse than with the out-the-window or helmet- mounted displays. However, the partial grid attachment condition appeared to improve hovering performance with the panel-mounted display.

Introduction

Current advances in aircraft technology and design have incurred a large information-assimilation requirement for the crewmembers of those aircraft. Therefore, it has become necessary to identify what information crewmembers actually use to perform various missions and mission tasks and to present such information so that crewmembers can obtain it as effeciently as possible.

Methods of information presentation have been approached in a variety of ways. The majority of research, however, has focused on the presentation of visual information with two-dimensional symbols or numbers that reflect individual dimensions of aircraft activity. For example, heading, range, vertical velocity and altitude are each represented by a separate symbol in a head-up display (HUD) while pitch and roll are almost always combined into a single representation (an artificial horizon).

Previous research has suggested that it is necessary to present certain information separately for unambiguous interpretation. On the other hand, greater information assimilation also has resulted when correlated information is integrated into in a single representation[1]. If more information is to be represented within a single symbol, it follows that this can be accomodated most effectively by using symbology with more than two dimensions; i.e. perspective or stereoscopic displays.

Research regarding the use of three-dimensional (3-D) symbology has been initiated in a number of areas. Grunwald[2] developed a tunnel-in-the-sky display (in 3-D) that integrates flight-path control information previously presented separately. The Super Cockpit[3] and the Virtual Environment Display System[4] use 3-D symbology in conjunction with stereoscopic displays.

In addition, basic research on the visual perception of motion has employed various types of 3-D symbology as ground texture. This research indicated that pilots use optical splay angles most effectively in detecting altitude changes when viewing displays passively[5] and optical texture-density most effectively when actively interacting with displays[6]. Since visual cues for the perception of motion are required for maintaining aircraft stability and altitude, performance on these tasks should be enhanced if a perspective display is presented in conjunction with a ground reference.

Two experiments were conducted in a fixed-base simulator to investigate various visual display alternatives. In the first experiment, a perspective display (grid pattern) was superimposed over computer-generated terrain. Subjects were required to maintain their initial altitude with displays consisting of only the ground, only the grid, or both. Four grid patterns were used: vertical lines that converged on infinity, i.e. in the center of the screen (meridian grid), horizontal lines that increased in density as they reached infinity (lateral grid), both vertical and horizontal lines as described above (square grid), and a quasi-random pattern of dots (dot matrix). Line drawings of the various grids are presented in Figure 1. The meridian grid pattern provided only optical splay cues, the lateral grid pattern provided vertical density cues and relative vertical position cues, a dot matrix pattern provided vertical density cues and weaker

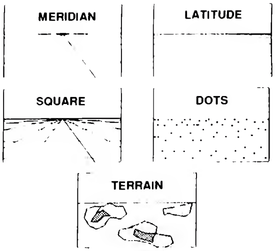


FIG 1. LINE DRAWINGS OF FOUR GRIDS AND TERRAIN

relative vertical position cues, and a square grid pattern provided vertical density cues and relative vertical position cues as well as optical splay cues. Previous part-task simulation research[6] would predict that patterns with relative vertical position cues and vertical density cues would be superior.

The second experiment was designed to identify the impact of a perspective grid on the performance of more complex flight-path control tasks. In addition, the dynamics of the grid with respect to the aircraft and the type of display used to present visual information were manipulated. Subjects were required to fly a specified groundtrack through a slalom course and perform a visual approach to a hover in a world with pylons or without pylons. The world with pylons is depicted in Figure 2. A square grid pattern (1000 ft x 1000 ft) followed movement of the aircraft with respect to pitch, roll, and yaw (complete grid attachment), or the grid remained stationary with respect to pitch, roll and yaw (partial grid attachment). The grid always followed aircraft movement with respect to forward, lateral and vertical translation. The three display types were out-the- window, panel-mounted and helmet-mounted. There were no a priori predictions made about the outcome of this experiment; it was exploratory in nature.

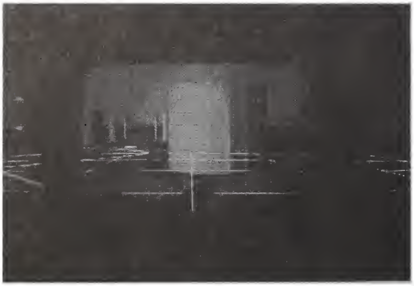


FIG 2. SCANNED IMAGE OF PYLON WORLD

Experiment 1

Method

Subjects. Five males served as voluntary participants; four of the subjects were paid for their participation. All subjects had or were corrected to 20/20 vision or better, were right-handed, and their ages ranged from 22 to 32. This information was obtained verbally from the subjects. Two of the subjects were rated pilots. One subject was excluded from the following analyses because his pattern of results was internally inconsistent and strikingly aberrant from the other four subjects. This suggested that task learning had not adequately stabilized prior to data collection.

Apparatus. The NASA Interchangeable Cab (ICAB) fixed-base simulator was used in a rotary-wing configuration. A three- window visual display was produced by a Singer-Link Digital Image Generator (DIG1) driven by a Perkin-Elmer computer. A Xerox Sigma 8 processor computed the flight dynamics, with the cyclic programmed to control vertical motion only.

Design and Procedure. The experiment was a 2x2x4 fully crossed repeated measures design with two control conditions. The independent variables were background, field-of-view (FOV), and grid type, respectively. The two types of background were completely black and textured ground with sky and a horizon line. The two types of FOV were three large windows (120 deg) and center window only (40 deg). The four grid types were latitude, meridian, square and dot matrix. The initial simulated altitude for all of the above conditions placed the subject 1000 ft above the grid; and for the conditions when a textured ground was present, the grid was placed 1000 ft above the ground. Thus, the subject's initial simulated altitude was 2000 ft above the ground when it was present.

The control condition contained no grid and, therefore, could be conducted only when a textured ground was present. However, it was not clear what the subject's initial simulated altitude should be for this condition. As stated above, the subject was always 2000 ft above the ground, but was only 1000 ft above the grid. Removing the grid placed the subject with an initial simulated altitude of 2000 ft above the ground. This placed the subject 2000 ft above his visual reference, whereas all other conditions placed him only 1000 ft above a visual reference (i.e., a grid). Therefore, to ensure an unambiguous comparison between the grid conditions and the control condition, two control conditions were conducted. The initial simulated altitude placed the subject 2000 ft above the ground for one control condition, and 1000 ft above the ground for the other control condition.

Subjects were seated in the cockpit and were given instructions on use of the cyclic for altitude control and the task to be performed. Subjects were required to maintain the altitude presented in the initial visual scene. Altitude (motion in the vertical plane) was the only flight parameter that could be controlled by the subject, although the illusion of motion was produced by simulated wind disturbances in the vertical, longitudinal and lateral planes.

Subjects were given approximately 10 min to familiarize themselves with control stick movements and the resulting altitude changes. During this period, all instruments were available for use by the subject. This training period was followed by 10 training trials. Training trials were identical to data trials in all ways except that the data collected was not included in the analysis. During all training and data trials, the flight instruments were not available to the subject for reference.

An individual trial began when the subject was seated in the simulator with one of the visual conditions displayed on the windscreen. The subject would press a button in the cockpit to initiate the trial. During the first ten sec of the trial, the subject viewed a static image of the visual condition to be maintained during the trial. At the completion of this ten sec interval, a tone sounded to alert the subject that wind disturbance and data collection would begin. At this point, the subject began to maintain altitude by moving the cyclic in response to perceived changes in altitude that occurred as the result of the wind disturbance. A second tone indicated the end of the trial. Elapsed time between the two tones was 220 sec. The controller provided performance feedback to the subject at the end of each trial.

Trials were run in sets of four or six. A set containing four trials included a trial with each of the grid configurations (latitude, meridian, square and dot matrix). A set containing six trials included a trial with each of the grid configurations plus two trials when no grid was displayed (the two control conditions). If the test condition was a black background or only the center window (40-deg FOV), the four-trial set was employed. The six-trial set was used only for the full-window (120-deg FOV) with textured ground condition.

Results

Median adjusted root mean square errors (ARMSE) for altitude were obtained as measures of altitude control ability. These values were calculated separately for each trial as follows. An ARMSE for altitude was calculated at a 25 Hz sampling rate over 10 sec. This was repeated for successive 10-sec intervals and resulted in 22 ARMSEs from a single trial (total trial time was 220 sec). Medians were calculated from the 22 ARMSEs.

Because there were unbalanced factors in the design (six trials for textured ground and four for black background), three separate repeated measures analyses of variance (ANOVAs) were conducted. Since this increases the possiblity of making a type 1 error, only significance levels of less than 0.01 were interpreted as a rejection of the null hypothesis.

The first ANOVA tested median ARMSE for altitude as a function of grid, background and FOV. A significant effect of grid on median ARMSE was present ($F(3,9)=19.05$, $p<0.001$) and can be viewed in Figure 3. The grids associated with the best performance were the latitude and square grids. The dot matrix and meridian grids were associated with the poorest altitude control performance. No significant effect for background was found, but the interaction between grid and background approached significance ($F(3,9)=4.90$, $p=0.028$). Variability in altitude control with the meridian grid decreased when textured ground was present. There was no significant difference between the two FOV conditions, nor did FOV interact with background or grid.

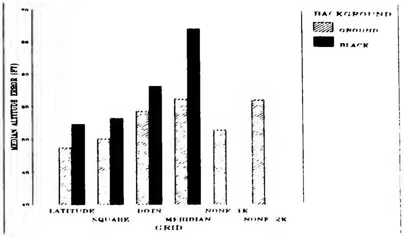


FIG 3. MEDIAN ALTITUDE ERROR AS A FUNCTION OF GRID AND BACKGROUND

The second ANOVA tested median ARMSE for altitude as a function of the grids with the textured ground and the two control conditions (textured ground without a grid). This can be seen by comparing the hatched bars of Figure 3. The control conditions represented altitude control at 1000 ft (none-1k) and 2000 ft (none-2k) above the ground. The differences due to grid persisted ($F(5,15=40.1075$, $p=0.002$). Performance on the 1000-ft control task was similar to that on the latitude and square grids (although the grids appeared to improve performance); whereas, performance on the 2000-ft control task was comparable to that on the dot matrix and meridian grids.

Finally, the third ANOVA tested median ARMSE for altitude as a function of the grids with the black background and the 1000-ft control task over the textured ground. Again, the differences due to grid are present ($F(4,12)=167.8790$, $p<0.001$). Performance with the textured ground was not significantly different from performance with the latitude and square grids. However, altitude control with the dot matrix and meridian grids was more variable than with the textured ground. This comparison is evident by viewing the black bars and the hatched none-1k bar in Figure 3.

Discussion

Comparisons with the control conditions indicated that performance with the latitude and square grids was comparable to performance at 1000 ft above the ground with no grid. This was true if the grids were over textured ground or if the background was black. These results suggest that subjects used the square and latitude grids to maintain altitude even when textured ground was also present, and that the cues provided by these grids were comparable to those provided by the textured ground alone.

Performance with the dot matrix and meridian grids, when textured ground was present, was comparable to performance at 2000 ft above the ground with no grid. When the meridian grid was presented with the black background, performance degraded even further. These results suggest that subjects did not use the dot matrix and meridian grids to maintain altitude when textured ground was also present. The meridian grid with the black background appeared to provide the least useful cues for altitude control. It was only in this condition that the subject was required to maintain altitude solely with optical splay information. Obviously, this suggests that optical splay cues are not the primary means of transmitting information about changes in altitude.

In summary, vertical cues obtained from latitude lines produced altitude control performance that was less variable than optical splay cues provided by meridian lines. This finding replicates the results of an earlier part-task simulation.[6] Since performance with the dot matrix grid was substantially below that with the square and latitude grids, it appears that vertical cues produced by the dot matrix grid were not comparable to those produced by vertical lines. It is possible that the total number of dots used in the present study was too sparse to provide useful vertical cues. The density of the dots was determined by the square grid pattern; a dot was provided for every intersection of the lateral and vertical lines. However, the dots were dispersed randomly. Therefore, it is also possible that the random arrangement of dots interfered with the subject's ability to obtain useful vertical cues. Further research would clarify this issue.

Experiment 2

Method

Subjects. Five helicopter-rated males served as voluntary participants; two of the subjects performed all of the display conditions, two of the subjects performed two of the three display conditions (out-the-window and panel-mounted) and one subject participated in only one of the display conditions (helmet-mounted only). All subjects had or were corrected to 20/20 vision or better and were right-handed. This information was obtained verbally from the subjects.

Apparatus. The apparatus was the same as Experiment 1 with the following exception: all aircraft controls were programmed for use by the pilot as they would be used in actual rotary wing flight (i.e., the subject controlled all axes of motion, not vertical motion only). In addition, the helmet-mounted display was a Honeywell Integrated Helmet and Display Sight System (IHADSS) and the panel-mounted display was a 13" diagonal CRT located directly below the center window of the out-the-window display. Visual scenes in the helmet- and panel-mounted displays simulated forward-looking infra-red (FLIR) imagery and therefore, was monochrome with a slight green tint. The IHADSS consisted of a monocular one-inch CRT and magnifying optics that projected the imagery onto a combiner lens. It was attached to the right side of the pilot's helmet such that the combiner lens was centered in front of the pilot's right eye.

Design and Procedure. The design of the experiment was a repeated measures 3x3x2 fully crossed factorial. The independent variables were display method, grid dynamics, and ground reference, respectively. The three display types were: (1) out- the-window (consisting of three large windows, 120 deg FOV), (2) panel-mounted (40 deg FOV with a visual image identical to the center window scene in the out-the-window display) and (3) helmet-mounted (the same visual presentation as the panel-mounted display). The grid was a 1000 ft x 1000 ft square grid pattern and the center of the grid always remained 50 ft below the center of gravity of the aircraft; thus, 500 ft of grid was available to the front, rear and sides of the aircraft. The three types of grid dynamics were: (1) no grid, (2) the grid followed the movements of the aircraft with respect to pitch, roll and yaw (complete attachment), and (3) the grid remained stationary with respect to pitch, roll and yaw of the aircraft (partial attachment). The grid always followed the movements of the aircraft with respect to forward, lateral and vertical translation. The flat, textured terrain was identical for both ground reference (visual world) conditions and contained a curved road simultaneously damped in both frequency and amplitude. However, in one condition, 1000 ft high cylinders of decreasing diameter were placed within the bend of each curve such that the distance from the center of the road to the center of the cylinder was always equal.

Pilots were given approximately 10 min to familiarize themselves with the flight dynamics model, the visual scene, and the two flight tasks. The two flight tasks were: to fly along the curved road (slalom course) and to descend to a 40-ft hover from 1500 ft (descent to hover). During this period, all instruments were available for use by the pilot. This training period was followed by four training trials. Training trials were identical to data trials in all ways except that the data collected was not included in the analysis. During all training and data trials, the flight instruments were not available to the pilot for reference. Training trials were conducted prior to the collection of data for each display condition. During the panel- mounted and the helmet-mounted display conditions, the three large windows in the simulator were baffled to remove glare.

An individual trial began when the pilot was seated in the simulator with one of the initial conditions displayed on the windscreen. The initial condition for the slalom course placed the aircraft 500 ft above the ground with a forward velocity of 70 knots. The initial condition for the descent to hover placed the subject 1500 ft above the ground with a forward velocity of 100 knots and at a ground distance of 6500 ft from the helipad.

When all of the trial and data collection conditions were ready, the pilot would press a button in the cockpit to inltiate the trial. The pilot was required to maintaln his altitude and groundspeed as he traversed the slalom course; and he was required to maintain a constant glideslope as he descended to a 40-foot hover over the helipad. The pilot ended a trial by pulling the weapons trigger on the cyclic when he believed he had completed the slalom course or stabilized at a 40-ft hover. The controller provided performance feedback to the pilot at the end of each slalom course trial.

Trials were run in sets of four. A set containing four trials included two slalom course trials and two descent to hover trials. Three sets were conducted using each display method. For a given pilot, the out-the-window display sets were completed first, followed by the panel-mounted display sets, and the helmet-mounted display sets. The three sets within a display method differed as a function of grid dynamics and were randomized for order of presentation.

Results

Since this study was exploratory in nature, a multitude of dependent measures were obtained and analyzed for each task. Due to limited space, only a subset of those measures will be presented in this paper. Separate ANOVAs were performed for each dependent variable and for each task. Data were sampled at a rate of 25 Hz during each trial and every third data point was used in the analysis. The data were further reduced by averaging the data points within 200-ft bins. For the slalom course, this resulted in six bins for each section; for the descent to hover task, it resulted in 27 to 31 bins. The number of bins differed as a function of ending position because on some trials, pilots stopped short of the helipad and on other trials, they overshot it. Data from the two tasks will be reported separately.

Slalom course. For analysis purposes, the task was divided into straight runs (the 1200 ft prior to the beginning of a turn) and turns (the 1200 ft during a turn, with the apex in the middle). Means and ARMSEs for path error and altitude were analyzed for each of the above. Unfortunately, slalom course data from the helmet-mounted display were not available for analysis prior to the required completion date of this paper; therefore, only results for the out-the-window and panel-mounted displays will be discussed for this task.

As might be expected, ARMSEs increased significantly for path error (F(2,6)=4.97, p=0.05) and altitude (F(2,6)=14.87, p=0.005) as the turn maneuver became more difficult to perform (i.e., as pylon diameter decreased). Both mean path error and altitude error were significantly greater when the panel-mounted display was used to regulate flight. The respective analyses yielded F(1,3)=22.31, p=0.02 for mean path error, and F(1,3)=11.34, p=0.04 for altitude error. In addition, there was a significant difference between the mean altitudes flown during the two display conditions (F(2,6)= 14.87, p=0.005). For the panel-mounted dispiay, the average was 589 ft ana for the out-the- window display, the average was 631 ft. There was no significant main effect or interaction for ARMSE for the three different grid conditions. There were no significant differences or interactions for the straight and curvilinear portions of the slalom course. See Figure 4 for a graphical representation of performance under the grid and display conditions for mean altitude.

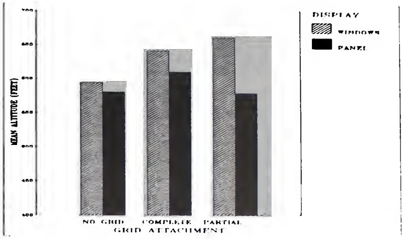


FIG 4. MEAN ALTITUDE AS A FUNCTION OF DISPLAY AND GRID

Descent to hover. The descent to hover task was separated into three phases for clearer interpretation of the data. The first phase was set-up for glideslope and descent rate; it encompassed 1600 ft and was not analyzed. The second phase was the descent, in which a stable glideslope and descent rate had been established; it encompassed 2200 ft. Mean glideslope was analyzed for these data to identify any differences across conditions. The third phase was the slow to a hover, in which nearly all of the dependent variables were transitioned to zero and altitude was to be maintained at 40 ft. This included the final 1600 ft of the flight. These results will not be discussed in the present paper. However, mean groundspeed and altitude at the conclusion of each trial were analyzed for differences across conditions.

Several missing values surfaced during the analysis of these data because there were an unequal number of subjects across display conditions (see Subjects above). The missing values of a given cell were replaced with the mean of the values within that cell.

For the descent phase, a significant display by grid interaction was observed for mean glideslope (F(4,16)=7.435, p=0.001). As can be seen in Figure 5, the steepest glideslope was maintained with the grid that was attached to aircraft movement in all translational and rotational axes (complete attachment) when combined with either the out-the-window or helmet-mounted display.

However, when the panel-mounted display was employed, the steepest glideslope was maintained with the grid that was attached to aircraft movement in all translational axes but no rotational axes (partial attachment).

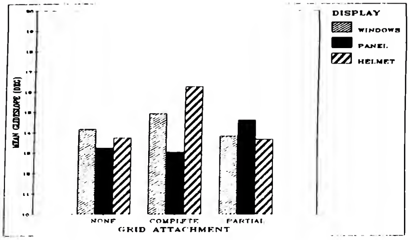


FIG 5. MEAN GLIDESLOPE AS A FUNCTION OF DISPLAY AND GRID

Data for mean altitude at the end of the trial are presented in Figure 6A. Significant differences were found for display (F(2,8)=35.077, p<0.001), grid attachment (F(2,8)=9.612, p=0.007) and a display by grid attachment interaction (F(4,16)=18.965, p<0.001) was found as well. Ending hover altitude was most accurate with the out-the-window display and least accurate with the panel-mounted display. However, performance on the latter appeared to improve when a grid was present. In addition, the partial grid attachment produced slightly better performance than the complete grid attachment. Performance with the out-the- window and helmet-mounted displays appeared unaffected by the three grid conditions.

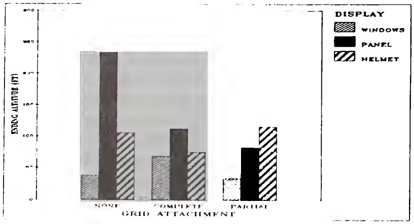


FIG 6A. MEAN ENDING ALTITUDE AS A FUNCTION OF DISPLAY AND GRID

A significant effect of display (F(2,8)=29.365, p<0.001) and a display by grid attachment interaction (F(4,16)=5.124, p=0.007) were evidenced for mean ending groundspeed (See Figure 6B). Ending groundspeed with the out-the-window display was clearly slower than with the helmet- or panel-mounted displays. However, the helmet-mounted display produced significantly better performance than the panel-mounted display in the no grid and complete grid conditions. There were no large performance differences across the three grid conditions for both the out- the-window and helmet-mounted displays. With the panel-mounted display, only the partial grid attachment improved performance on ending groundspeed.

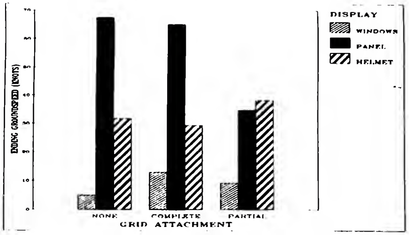


FIG 6B. MEAN ENDING GROUNDSPEED AS A FUNCTION OF DISPLAY AND GRID

Discussion

It is clear from the data on both tasks that flight performance with the panel-mounted display was significantly worse than with the out-the-window display. During the slalom course, pilots flew lower with the panel-mounted display than they flew with the out-the-window display. A possible explanation for these findings is the image minification produced by the panel- mounted display. Although the FOV of the panel-mounted display was 40 deg, there was a significant minification of the image (0.4x) due to the viewing distance. A smaller image suggests a higher altitude to the viewer; thus, the pilot would tend to decrease altitude. Such minification has been shown in other studies to have a significant effect on depth judgments during flight.[7] Comparisons of these data with the helmet-mounted display data may shed some light on this possibility because the viewing distance with the helmet-mounted display was conformal and, therefore, no image minification occurred.

It should be noted, however, that mean ending altitude for the descent to hover task was higher for the panel-mounted display than for the other two displays, and that, during the slalom course, pilots flew higher than the target altitude with both displays (out-the-window and panel-mounted). The altitude control required during the descent to hover task is dynamically different from the altitude control required for the slalom task. In the descent to hover, the pilot must maintain a constant change in altitude, whereas, in the slalom course, the pilots were required to maintain a constant altitude. Therefore, the resulting minification for the descent to hover task may have interacted with optical flow in a non-linear fashion as compared to the slalom course. With regard to the target altitude for the slalom course, it is possible that the pilots lost track of the target altitude and adopted a different reference altitude.

Although there were a number of differences among the three displays (such as image minification), two essential differences were: FOV and field of regard (FOR). The out-the-window display had a large FOV and FOR; the panel-mounted display had a small FOV and FOR, and the helmet-mounted display had a small FOV and a large FOR. As would be expected, the best overall performance occurred when the FOV and FOR were both large (out-the-window display). Performance was degraded when the FOV was restricted but FOR was not (helmet-mounted display), and it was further degraded when both FOV and FOR were restricted (panel-mounted display).

It appears that performance was comparable for the helmet-mounted and out-the-window displays with regard to grid attachment conditions. The steepest glideslopes on descent were maintained with the complete grid attachment when the helmet-mounted and out-the-window displays were employed, and ending altitude and groundspeed showed no large differences between the no-grid and grid conditions for these displays. On the other hand, the steepest glideslope was maintained for the panel-mounted display with the partial grid attachment, and large differences between the no-grid condition and the grid conditions were found for this display. These findings suggest that the addition of a grid improves performance that is degraded when FOR is restricted rather than when FOV is restricted. In particular, the grid with partial attachment to aircraft movement provides the most comprehensive improvement in performance. However, performance with the restricted FOV displays (panel- and helmet-mounted displays) never reaches the level of the wide FOV display (out- the-window).

In summary, the grid attached to aircraft movement in translational, but not rotational, axes may provide some of the visual cues for altitude and groundspeed control that are removed when FOR is reduced. Since this grid attachment condition essentially moves the ground reference plane closer to the pilots, smaller changes in pitch, roll and yaw should be noticeable. In turn, this should allow them to better control the aircraft. The complete grid condition, on the other hand, is essentially an extension of the aircraft. Using the horizon as a reference plane, this grid condition should enhance the perception of pitch and roll changes, but not necessarily changes in yaw. Perhaps it is this difference in the treatment of visual cues for yaw that supports the aforementioned findings. It should be remembered, however, that there was some amount of performance decrement (perhaps due to decreased FOV) that was not improved by the presence of a perspective grid during the descent to hover task. As always, further research is required, particularly into the visual cues removed when FOV and/or FOR is restricted.

Acknowledgments

Several people have made important contributions to this paper. We sincerely thank them for their efforts. They include Anil V. Phatak, Paula Bray and Stephanie Hinz.

References

1. Garner, W.R. The Processing of Information and Structures, Hillsdale, NJ: Lawrence Erlbaum Associates, 1974.
2. Grunwald, A.J. Tunnel display for four-dimensional fixed-wing aircraft approaches. Journal of Guidance, Control and Dynamics Vol. 7, May-June, 1984, 369-377.
3. Furness, T.A. III The Super Cockpit and its Human Factors Challenges. In Proceedings of the Human Factors Society 30th Annual Meeting, Dayton, OH, September 29-October 3, 1986.
4. Fisher, S., McGreevy, M., Humphries, J. and Robinett, W. Virtual Environment Display System. ACM 1986 Workshop on Interactive 3-D Graphics, Chapel Hill, NC, October 23-24.
5. Wolpert, L., Owen, D. and Warren, R. The isolation of optical information and its metrics for the detection of descent. In D. H. Owen (Ed.) Optical flow and texture variables useful in simulating self motion (II), Columbus, OH: The Ohio State University Aviation Psychology Laboratory, June, 1983.
6. Johnson, W.W., Tsang, P.S., Bennett, C.T. and Phatak, A.V. The visual control of simulated altitude. In Proceedings of the Fourth International Symposium on Aviation Psychology, Columbus, OH, April 27-30, 1987.
7. Randle, R.J., Roscoe, S.N., Petit, J.C., "Effects of magnification and visual accommodation on aim point estimation in simulated landings with real and virtual displays," NASA TP 1635, 1980.

88-4636-CP

EYE-SLAVED AREA-OF-INTEREST DISPLAY SYSTEMS: DEMONSTRATED FEASIBLE IN THE LABORATORY

G. Blair Browder
Walt S. Chambers
Advanced Simulation Concepts Division
Naval Training Systems Center
Orlando, Fl.

Abstract

Two unique simulator display concepts that use eye-slaved, area-of-interest (AOI) technology have been evaluated for their engineering and psychophysical characteristics. These visual systems exploit the limitations of the physiology of the human eye. That is, that the eye perceives the world as high resolution everywhere even though the high acuity region of the eye is less than 10 degrees wide in the direction of gaze. The results of these evaluations prove that this technology is both feasible and practical, and can potentially save millions of dollars in visual simulators where high detail and resolution approaching that of the human eye is required over a large field-of-view.

Introduction

The use of visual displays with an area-of-interest (AOI) inset that is target-slaved is a common solution to the design of simulators that have a requirement for a visual display exhibiting high resolution and detail. However these visual displays are restricted to high resolution only in the designated target area. Visuals with eye-slaved AOI displays on the other hand provide high resolution wherever the observer is looking, giving the observer the perception that high resolution exists everywhere in the total field-of-view.

The Navy and Air Force have evaluated two unique visual test beds that use the eye-slaved AOI display concept and have found through extensive formal experiments and demonstrations that the concept is feasible and should advance to an engineering prototype design.

One of these display systems is the ESPRIT (Eye-Slaved Projector Raster Inset) visual test bed, an Independent Research and Development (IRAD) program of the Link Flight Simulation Division of the Singer Company [1], [2]. This visual test bed, located at Binghamton, N.Y., has been extensively evaluated over the last few years, including both engineering tests and psychophysical evaluations. The ESPRIT is characterized by the use of 'off-the-head projection systems' and uses conventional color light-valve projectors.

The other display concept was evaluated on the VDRT (Visual Display Research Tool) test bed located at the Navy's VTRS (Visual Technology Research Simulator) in Orlando, Fl. [3], [4], [5]. The VDRT display concept differs from that of the ESPRIT primarily in that it's projection system is mounted on the helmet of the observer. A full-color laser display system with fiber-optic cables transports a modulated laser beam up to the helmet. VDRT tests and experiments have been conducted over the past year to demonstrate the feasibility of this concept. At this writing, results are available from the first of a series of experiments planned in the total VDRT evaluation.

It is emphasized that although the VDRT and the ESPRIT differ widely in the way they each achieve their goals, conceptually they are based on the same characteristics of the eye. Both visual displays provide the observer with a perception of high resolution over the entire field-of-view by inserting an eye-slaved, high resolution AOI display into a much larger field-of-view background display with much lower resolution.

The ESPRIT Evaluations

The ESPRIT visual test bed evaluations started early in 1984 with a system that was subsequently (in early 1985) improved in many areas for further evaluation. The first visual test bed was composed of the following subsystems:

1) Observer station with basic stick and throttle controls
2) Fixed background projector (GE monochrome light-valve)
3) Foveal projector with azimuth, elevation, focus, zoom, and brightness controls (GE monochrome light-valve)
4) Flat display screen (field-of-view 76x64 deg)
5) Digital image generator (Link's DIG I)
6) Foveal and peripheral electronics
7) Helmet with integrated head and eye tracking systems

The foveal projector, located near the observer, projected a round AOI with an intensity-feathered outer region. The AOI display, slaved to stay aligned with the eye's gaze direction, was projected into a "video hole" cut into the fixed background. Nominally, the AOI was 18 degress in diameter including a 3 degree-wide blend ring. The fixed background projector was located above and behind the observer on a mezzanine deck. Figure 1 shows a view of the visual test bed and display from a point above and behind the observer.

Engineering Tests

Before starting the psychophysical evaluations of the ESPRIT visual test bed, engineering tests were conducted on the complete test bed to characterize and define the visual display both statically and dynamically. The visual display was tested to define traditional visual performance parameters such as brightness, contrast, resolution, and distortion. Special test hardware was devised to measure transport delays from onset of head and eye motion to first perceived display response. Also, response time from onset of eye saccade to response of the foveal projector servo was determined. Some of the more important visual test bed display characteristics are summarized below:

1) Nominal AOI size = 18 degrees (circular)
2) Nominal blend-ring size = 3 degrees radius
3) Background field-of-view = 76X64 degrees
4) Average brightness = 1.6 foot-Lamberts (color)
5) Resolution of AOI = 2.5 arc minutes (20% MTF)
6) Resolution of background = 11 arc minutes (20% MTF)

Psychophysical Evaluation - Phase I

The first experiments were conducted using 20 observers in a mix of pilots and non pilots. A special database was constructed to provide both a tracking task and a target detection/identification task. The database featured a long canyon with a narrow path on the floor for tracking tasks and randomly placed targets and decoys along the canyon walls for target detection and identification tasks. The walls and the floor were both textured with a checkerboard pattern, as illustrated in Figure 1.

Task I was designed to evaluate the impact of the AOI presence or absence on tracking performance (maintaining position over the narrow canyon path). During this evaluation, each subject was trained thoroughly in the task of "flying" down the canyon. Altitude and forward speed were both fixed. Lateral control was achieved with the joystick. A drift factor was introduced to add some difficulty to the task. Two levels of background resolution were used in this evaluation phase: 18 arc minutes (achieved by defocusing) and a nominal 11 arc minutes pixel spacing. The experimental design for Task I is shown in Table 1.

The following summarizes Task I results:

1) Use of the AOI in the background display resulted in practically the same tracking performance as the case of background display only.
2) Background display resolution at the two levels used made no appreciable difference on tracking performance.

The experiments of Task II were aimed at evaluating two important visual display design parameters:

1) AOI size requirement.
2) Maximum transport delay allowable between the onset of head/eye movement and the resultant display response.

'AOI size requirement' impacts display resolution capacity, while the 'maximun transport delay allowable between head/eye onset and display response' defines how much computational time is available (time required to generate the image is the major contributor).

The scope of Task II was increased to include detection of small targets located on the canyon walls while at the same time performing the basic tracking task. Task II observer performance was evaluated for AOI diameters of 12, 18, and 28 degrees. With the AOI size set to a nominal value of 18 degrees, additional transport delay was introduced such that the total delay extended from about 140 to 290 milliseconds. The experimental design for Task II is presented in Table 2.

Task II results were encouraging in that little performance variation was observed with the AOI size variation or with the extra transport delay until the total system transport delay exceeded 190 milliseconds. Beyond this level, performance degraded appreciably.

Phychophysical Evaluation - Phase II

Phase II experiments, designed to study AOI requirements for size and brightness, were conducted on a much-improved visual test bed. The specific improvements made were:

1) Monochrome projectors replaced with full-color GE light-valve projectors.
2) Oculometer digital processor upgraded to a system that provided more processing time thus eliminating timing-out.
3) Helmet-fit improved with a new four-bladder design for a more secure fit.
4) Small cockpit shell with F-16 profile and HUD outline added to observer station to add realism and limit observer's FOV to that of the screen (76 x 64 degrees).

Twenty-six observers with a mix of pilots and non-pilots were selected to perform experiments to study the effect of AOI diameter, blend region diameter, and AOI-to-background brightness ratio. Also, the background update rate in the digital image generator was introduced as a parameter, with values of 30 Hz and 60 Hz investigated. The AOI is always updated at 60 Hz. The possibility of using a lower background display update rate is important since it would imply that more time could be allocated for image generator computation. The size of the AOI was varied from 10 to 28 degrees (including blend ring), while the blend ring itself was varied from 2 to 6 degrees. Resolution inside the AOI was maintained at about 4.8 arc minutes for all combinations of AOI and blend ring size. Two levels of AOI to background brightness ratio were studied, 1.5 : 1. and 2.0 : 1.

The experiment basically followed the same format as Phase 1, with an initial training period of at least 10 runs with the tracking task, followed by 5 runs for each condition studied with the observer performing both the tracking and target identification tasks. Observers were interviewed after each condition to record their subjective opinions including any sickness tendencies. A 5 run set typically required about 5 minutes.

Briefly summarizing the results, the higher brightness ratios had little impact on performance with the exception of slightly better target identification. In general, little performance variation was noted for any of the conditions, including background rate update, AOI size, and blend-ring size. It is important to note that some facts are not borne out by the performance data. For example, observers prefer the larger AOI and blend rings in spite of the flat performance curves. Observers also stated that the AOI region appeared to be invisible when the task load reached a peak (during the faster moving part of each canyon run). General comments were that they liked the visual display system.

Psychophysical Evaluation - Phase III

During the Phase III experiments, two unique tests were devised, both aimed at determining more about the display system when the observer is confronted with tasks demanding large and rapid eye movements.

Saccadic Test

First, a saccadic (rapid eye movement) test was designed to determine an observer's ability to locate a target, saccade quickly and accurately to that target, identify the target, and then rapidly continue to saccade to the next target. Each target, if properly identified, would provide information needed for the next saccade. This experiment was first performed using a hardware device (representing the real world), and then the experiment was replicated using the ESPRIT visual test bed with the appropriate databases.

To represent the 'real world', a square frame with a computer-controlled light-emitting diode configuration at each corner was constructed (see Figure 2). At each corner target, three short radial lines pointed toward the other three corner targets. Only one radial line at a time could be lighted, indicating to the observer which corner was the target to be saccaded to next. At the same time, this square frame was also replicated for repetition of the experiment using the visual test bed.

The saccadic test started by having the observer look at a specific corner, start saccading as instructed by the radial line direction, and to saccade from target to target as fast as he could. The observer was instructed to continue a saccade even if he realized he had made a wrong identification. Twenty-six observers, again a mix of pilots and non pilots, were used in the experiments. Variables investigated were:

1) Size of the square (displayed frame only)
2) Extra transport delay introduced into the display response
3) Flat-colored background versus one with texture in the displayed frame
4) Background only vs background with the AOI inset

Variables recorded were time-to-saccade to a target, fixation time on each target, and number of correct and incorrect target recognitions.

The most important result was that observer performance was essentially the same when the replicated display frame was used as was the case when the actual hardware frame was used. When the AOI display was used, observer performance was clearly superior, with 40% more target recognition errors made when the AOI was not used (background display only). The additional transport delay (up to 50 milliseconds) had only a minor effect on accuracy, but did reduce the number of targets identified. The larger frame size caused longer saccade times as expected. No significant difference was noted due to the presence or absence of background texture.

Focal-Ambient Test

It is widely recognized that one's visual system has two functional modes: the focal mode and the ambient mode [6]. The focal mode serves the tasks requiring object discrimination and identification, whereas the ambient mode serves those tasks involving locomotion and orientation. The object of this test was to determine if the extensive use of the eye-slaved AOI in target identification tasks degrades or affects performance of tasks requiring the ambient mode.

For the focal-ambient experiments, two basic tasks were defined. One task required the observer to control his altitude (only) while 'flying' down the center of the canyon under conditions of low altitude and high speed. The other task required the observer to detect and identify targets located at the end of the canyon. Targets were triangles with the apex truncated while decoys were triangles without the truncation. These targets were sized so that they could not be identified without the high resolving power provided by the AOI.

Figure 3 shows a typical observer's view looking down the canyon. Three texture densities were used along the 100 foot-wide floor and walls. Starting at an altitude of 50 feet, forward speed was maintained at three optical flow rates after a brief acceleration. An altitude control bias was introduced as a disturbance for all runs, requiring corrective control inputs by the observer during the entire run. Field-of-view was also varied for the background to determine the importance of peripheral stimulation on the altitude control task. The conditions and their values used are as follows:

1) Target detection task - present or absent
2) AOI - present or absent
3) Field-of-view - 30, 45, or 60 degrees
4) Altitude control bias - 1, 2, or 4 feet per second
5) Forward velocity - 50, 100, or 200 feet per second
6) Texture density - 1, 2, or 4 ground units per eyeheight

Since it was not practical to do a complete factorial crossing of all conditions, only two factors were fully crossed, the presence or absence of the target detection task and the presence or absence of the AOI. A total of 26 observers were used in the experiment. Each observer flew the thirty-six combinations shown in Table 3 in a random order. The sessions were then repeated twice more on successive days.

Basically, the experiment showed that the ambient task of altitude control was affected very little by the presence vs absence of the AOI. However, altitude control was found to be affected somewhat when the simultaneous task of target detection was added. Results of this dependence are inconclusive, however, since the affect of altitude control was both positive and negative. Altitude control was positively affected in that altitude RMS error was less and was negatively affected in that response time to first altitude correction was increased. The wider field-of-view case showed somewhat better and faster early corrective altitude control. But the bottom line is that the target detection task was impossible without the high resolution provided by the AOI, and the use of the AOI did not degrade the altitude control task performance.

The VDRT Evaluations

The first series of experiments directed towards demonstrating feasibility of the eye-slaved display concept used in the VDRT have been completed. These first experiments were designed to replicate experiments conducted a few years ago in the same dome/cockpit but with a display with a fixed projector. All other factors were the same (image generator, databases, aerodynamics, etc.). With Navy pilots flying the VDRT with it's eye-slaved AOI display, performance comparisons were made for the same tasks performed in the fixed projector configuration.

The VDRT visual test bed is composed of the following major subsystems:

1) T2-C cockpit with instruments and controls
2) Full-color laser-raster display system
3) Digital image generator (GE CompuScene I)
4) 20 foot diameter dome with high-gain retro-reflective screen surface
5) Helmet with projector/laser-scan optics and integrated head/eye-tracking systems

The VDRT is unique in that it projects from the helmet onto a dome an image that stays centered with the eye's gaze direction. A full-color laser raster system transmits line images over fiber-optic cables up to the helmet optics. Head and eye-tracking systems, mounted on the helmet, provide instantaneous gaze direction. The display image is composed of a relatively low resolution surround (referred to as the instantaneous field-of-view or IFOV), with a very high resolution AOI in the center of the IFOV. Both the IFOV and AOI are intensity-feathered in a small region to blend them into each other. Thus, when the image is centered with the eye's gaze direction, the resolution profiles of the human eye and the image are similar. The VDRT display is characterized by the following:

1) IFOV size - 140 x 100 degrees
2) AOI size - 27x24 degrees
3) Blend ring size - 5 degrees
4) Brightness - 10 foot-Lamberts
5) Resolution (AOI) - approx 3.5 arc minutes
6) Resolution (IFOV) - approx 18 arc minutes
7) Screen gain - at least 100 at peak
8) Display update - 60 Hz (all subsystems)

Due to the fact that the projection pupil and observer's eye-point are located near the center of the dome, and that the dome has a high-gain/highly directive surface, the observer perceives a bright, high-resolution image wherever he looks in the dome. Figure 4 shows the helmet/optics with fiber-optic cables that relay a line-image from the laser to the helmet optics.

Psychophysical Evaluations - First Formal VDRT Experiments

The VDRT has been successfully demonstrated to and flown by many Government agencies and contractors for more than a year. In the spring of 1988, the first formal experiments were completed. Those experiments were designed to replicate experiments (based on a 30 degree dive bomb task) made a few years ago on the same simulator with a fixed-projector display. The fixed background display was aided by a target-slaved AOI to provide very high resolution to permit easy detection of the target site. Based on preliminary results from those experiments, Navy pilots using the eye-slaved VDRT performed at least as well as on the visual with a fixed display system and showed no adverse behavior.

The experiments were conducted using 9 Navy pilots with considerable attack aircraft experience. The image generator and database used were the same, and thus provided the same scene content as the old projection system. The task replicated the 30 degree dive bomb runs, using the same initial conditions and same gunsight model. Performance parameters recorded for analyses were the aircraft speed and position at the start of the bombing run, flight path accuracy along the dive slope, aircraft speed and position at the bomb release point, and bomb impact accuracy.

The 9 Navy pilots flew a total of 64 bombing runs, 32 against a "gunnery-range" as target and 32 against a selected "building site" located close to other buildings as target. For half of the bombing runs, the pilots flew the simulator with the display in a head-tracked mode. The remaining half of the bombing runs were then repeated with the display in an eye-tracked mode.

At this writing, preliminary performance data indicates results that are very positive with respect to the eye-tracked concept used in the VDRT. Pilot performance was essentially the same for the dive bombing tasks replicated on the VDRT as pilot performance recorded back in 1983 using the fixed projector display with target-directed AOI. Also of interest is the result that pilot performance using the head-tracked display was essentially the same as pilot performance using the eye-tracked display. It is thought that this was due largely to the fact that the dive bombing task does not require a large field-of-view and thus does not require the large, rapid eye saccades associated with tasks such as air-to-air combat.

The next planned experiment will address air-to-air tasks which will require peripheral detection and identification of air targets while simultaneously flying formation. These tasks will demand more of what the AOI is designed to provide: high detail and high resolution over a wide field-of-view.

Conclusions

Results from extensive experiments and demonstrations conducted on the ESPRIT visual test bed and on the VDRT visual test bed provide convincing evidence of the eye-slaved display concept feasibility. The ESPRIT performance has been demonstrated to be practically unaffected by reasonable tolerance variations on important design parameters such as AOI size and transport delay in the system. It is also an important finding that simulator sickness incidences were almost non existent in both the ESPRIT and VDRT visual test beds.

It is concluded that eye-slaved visual display technology, whether helmet-projected or projected from a fixed site, has reached a demonstrated technology performance level to be advanced to a full-scale engineering development level to achieve the improvements in packaging, operation, and reliability required to transition this concept into the training community. One such development has been initiated by the British Ministry of Defense, under contract to Singer Link-Miles, to develop the Harrier, GR-5 simulator. The GR-5 simulator design guidelines will be a direct result of the evaluations on Link's ESPRIT visual test bed.

References

1. Tong, H. M.; Fisher, R. A., "Progress Report on an Eye-Slaved Area-of-Interest Visual Display," IMAGE III Conference, May 1984

2. Turner, J. A.; "Evaluation of an Eye-Slaved Area-of-Interest Display for Tactical Combat Simulation," 6th I/ITEC Conference Proceedings, October 1984

3. Breglia, D.; Spooner, A. M.; Lobb, D., "Helmet-Mounted Laser Projector," Proceeding of IMAGE II Conference, June 1981.

4. Breglia, D., "Helmet-Mounted Laser Projector, "Proceeding of the Third Interservice /Industry Training Equipment Conference, December 1981.

5. Breglia, D.; Windyka, D.; Barber, B., "VDRT - Performance Design Goals," Proceedings of the Seventh Interservice/Industry Training Equipment Conference, November 1985.

6. Leibowitz, H. W.; Post, R. B., "The Two Modes of Processing Concept and Some Implications," Organization and Representation in Perception, 1982.

Table 1 Task I Experimental Design

	TRAIN TO LEVEL PERFORM	TEST I 10 RUNS	TEST II 10 RUNS	RETEST 10 RUNS
10 SUBJ	LO RESOLUTION BACKGROUND 18 ARC MIN NO AOI	HI RES BACKGROUND 11 ARC MIN WITH AOI	HI RES BACKGROUND 11 ARC MIN NO AOI	LO RES BACKGROUND 18 ARC MIN NO AOI
10 SUBJ	LO RES BACK NO AOI	HI RES BACK NO AOI	HI RES BACK WITH AOI	LO RES BACK NO AOI

Table 2 Task II Experimental Design

	5 RUNS	5 RUNS	5 RUNS	5 RUNS	20 RUNS	5 RUNS	5 RUNS
10 SUBJ	NO AOI	NOM 18 DEG AOI	12 DEG AOI	28 DEG AOI	140-290 MSEC TRANSPORT DELAY/RANDOM	NOM 18 DEG AOI	NO AOI
10 SUBJ	NO AOI	NOM 18 DEG AOI	28 DEG AOI	12 DEG AOI	140-290 MSEC TRANSPORT DELAY/RANDOM	NOM 18 DEG AOI	NO AOI

Table 3 Conditions Selected for Focal/Ambient Test

EVENT NO.	TARGET DETECTION TASK	AOI	FIELD OF VIEW	CONTROL BIAS	FORWARD VELOCITY	TEXTURE DENSITY
1	PRESENT	WITH	60 DEG	2 FT/S	100 FT/S	2 GROUND
2	PRESENT	WITH	45	2	100	2 UNITS
3	PRESENT	WITH	30	2	100	2 PER EYEHT
4	PRESENT	WITH	60	4	100	2
5	PRESENT	WITH	60	1	100	2
6	PRESENT	WITH	60	2	200	2
7	PRESENT	WITH	60	2	50	2
8	PRESENT	WITH	60	2	100	4
9	PRESENT	WITH	60	2	100	1
10	ABSENT	WITH	60	2	100	2
11	ABSENT	WITH	45	2	100	2
12	ABSENT	WITH	30	2	100	2
13	ABSENT	WITH	60	4	100	2
14	ABSENT	WITH	60	1	100	2
15	ABSENT	WITH	60	2	200	2
16	ABSENT	WITH	60	2	50	2
17	ABSENT	WITH	60	2	100	4
18	ABSENT	WITH	60	2	100	1
19	PRESENT	W/O	60	2	100	2
20	PRESENT	W/O	45	2	100	2
21	PRESENT	W/O	30	2	100	2
22	PRESENT	W/O	60	4	100	2
23	PRESENT	W/O	60	1	100	2
24	PRESENT	W/O	60	2	200	2
25	PRESENT	W/O	60	2	50	2
26	PRESENT	W/O	60	2	100	4
27	PRESENT	W/O	60	2	100	1
28	ABSENT	W/O	60	2	100	2
29	ABSENT	W/O	45	2	100	2
30	ABSENT	W/O	30	2	100	2
31	ABSENT	W/O	60	4	100	2
32	ABSENT	W/O	60	1	100	2
33	ABSENT	W/O	60	2	200	2
34	ABSENT	W/O	60	2	50	2
35	ABSENT	W/O	60	2	100	4
36	ABSENT	W/O	60	2	100	1

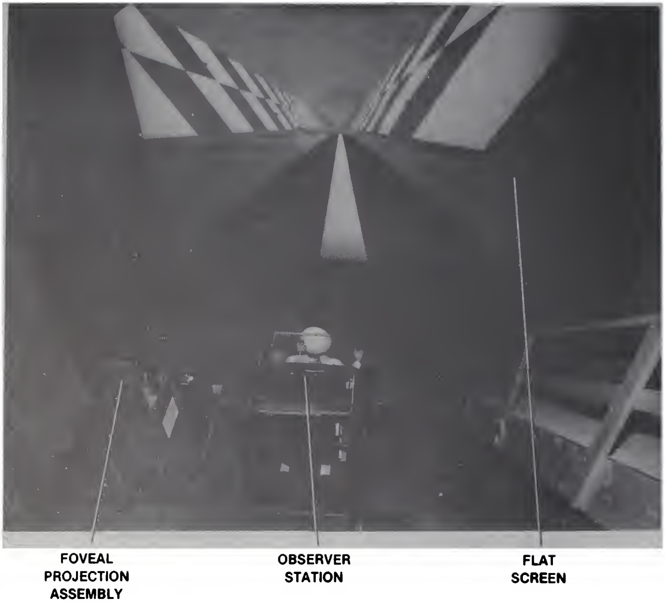


Figure 1 Eye-Slaved Projector Raster Inset (ESPRIT) Visual Test Bed

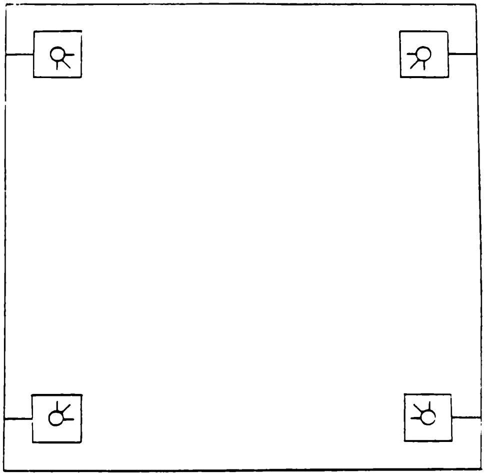


Figure 2 Frame Configuration for the Saccadic Test

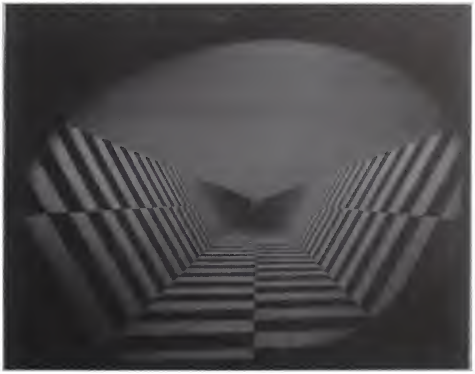


Figure 3 Canyon with Textured Floor and Walls

Figure 4 Helmet-Mounted Laser Projector (HMLP) Visual Test Bed

SOFTWARE TOOLS FOR BUILDING DEDICATED, REALTIME APPLICATIONS

B. Cothran and D. Comstock
Digital Equipment Corp.
Maynard Mass.

ABSTRACT

In today's cost critical environment, the design, development, and maintenance of software used in realtime simulators demand the use of sophisticated software tools. This discussion will describe, qualitatively and quantitatively, a set of software tools used for building dedicated, realtime applications. The multiprocessing, multitasking features, predictable time critical response capabilities, small/efficient realtime kernel, support of high level languages (ADA, Pascal, FORTRAN, and C) are essential features needed for a realtime operating environment.

The discussion will give a qualitative overview describing how a single computer architecture, with the use of a sophisticated set of software tools can be used to build realtime applications requiring single and or multiprocessing systems. The talk will provide specific technical information describing the use of a set of software tools in a typical program development and support life cycle. Software development and configuration control tools will be described as well, along with why these features provide life cycle cost savings. Standard product features such as ADA TM support, remote debugger, and performance analysis utilities will be discussed. Specific performance numbers will be provided supporting the discussions.

Characteristics of a realtime system include such requirements as, the need for low memory overhead, Low System and I/O Overhead (systems must be fast), Responsive to external, time critical events, predictability, dedicated and or Embedded applications; where most hardware systems are dedicated to a specific application, and the environment created is stable. In addition, where cost and time of development are critical considerations, development and execution environments become key. Addressing these issues, development becomes less cumbersome with the use of development tools available with general purpose operating systems, combined with ones ability to use high level languages for application coding. Add to this a toolkit, specific to the building of realtime application and creation of a run time environment, and you poses the ability to cut costs, increase productivity, and decrease your time to market.

HISTORICAL

The methodology for designing and developing realtime applications is changing. A large number of real-time programmers started their careers in the 70s with PDP-11s or similar designed machines. Some of the characteristics of development for those machines included:

- o 16-bit architecture.
- o no re-entrant code.
- o development and application on the same system.
- o Difficulty in moving a application from a single computer system to multiple computer systems.

The trends that we are currently observing are very different from those good-old-days.

- o 32-bit machines are being used for the majority of new applications.
- o Re-entrant code is the norm.
- o Distributed processors being used in order to obtain the processing power and I/O bandwidth needed for today's demanding applications.
- o Software development no longer being regarded as a art-form. It is now a field of engineering, with, often, large teams of engineers working on large computer systems configured, strictly, for only software development. These have all the the necessary peripherals and software development tools to aid in the business of manufacturing software.

DEVELOPMENT TOOLS

More and More we are seeing the need of Engineering Tools, both high level and low level tools. Examples of low level tools are utilities that we are most familiar with, -- editors, good compilers, linkers, and debuggers.

Higher level tools, that we are beginning to see people using, are utilities like CMS/MMS, LSE, and EPA. CMS/MMS is a set of utilities that control code modules for large applications. With large teams of engineers working on projects, we see the need for utilities that can build module libraries, allow for sets of modules to be part of application baselevels, and most importantly, show changes from module version to module version.

Another high level tool is LSE, an editor that understand the application language. This editor can give on-line help with language constructs, compile the edited module, and pinpoint compiled errors, all without the engineer having to leave the editor or go to the line printer for a module listing.

EPA is an example of a profiling and performance analyzer which helps an engineer to test his application and confirm that all modules have executed. EPA, can also, indicate where the application is spending it's time. This can allow the engineer to tweak portions of the application, to achieve higher throughput.

VAXELN

Another tool, that we are beginning to see, is the need for a tool to build dedicated, realtime applications -- VAXELN. VAXELN is a development tool, it is not an operating system. The runtime kernel has many of the attributes associated with an operating system, but we consider it to be a comprehensive tool for building realtime applications. VAXELN is a software toolkit that run on a VAX VMS system. The output of the toolkit is a standalone application running on a dedicated VAX computer.

All design and development with ELN can be done with high level languages. Just as many operating systems ship a macro assembler, we ship the Epascal compiler with the VAXELN toolkit. In addition to Epascal, VAXELN supports applications written in VAX C, VAX FORTRAN, and, optionally, VAX ADA. Drivers can be written in Epascal, VAX C, or VAX ADA. The VAXELN/ADA software allows applications to be written in VAX ADA. VAXELN/ADA is a fully validated implementation of the ADA language and compiles with ANSI-MIL-STD-1815-A-1983.

The application development is very similar to development on general purpose systems. The code has been developed either using LSE or another editor, written to a file. That file is then run through a compiler (Epascal, C, ADA or FORTRAN compiler). Out of the compiler comes a .obj file which has the same 'OBJ' structure common to VMS. That '.OBJ' file is then combined with other files using the standard VMS linker to produce a VMS '.EXE' file.

This '.EXE' file is combined with other previously developed '.EXE' files through the VAXELN System Builder along with DIGITAL-supplied VAXELN services, runtime libraries, VAXELN kernel, and I/O drivers, supplied by Digital or written by the user, to produce a VAXELN system file. This file represents a target application image that can be transported to a VAX target computer.

The VAXELN Debug utility is a extremely powerful set of programs which allow a user to debug multiple process or jobs, which comprise his VAXELN application. When the target is linked to the development system using Ethernet, the normal debug session is done at the VMS user's terminal. The VMS Edebug program links across to the ELN Debug module running on the target machine, or machines. This allows a user to set breakpoints, interrogate variables, start/stop jobs and do all the normal types of debug actions. The 'neat' attribute of Edebug is that multiple jobs can be debugged simultaneously, even if these jobs are on different VAX processors. For those people who are not using Ethernet or RTA and have stand alone systems, VAXELN has a local debugger that run from the target system's console terminal.

MULTIPROCESSING SUPPORT

Lets look at developing an application system. Suppose we have a typical application which is composed of jobs Alpha, Beta, Gamma and Fred. Applications tend to grown over time, through maintenance. Many times we find that the application will outgrow the target processor. The normal solution is to purchase a newer, higher performance computer, with all the associated costs for upgrading to a new machine. However, with VAXELN, if the design has been done correctly, those programs can be divided across multiple processors. So, rather than going off and purchasing a new larger processor and trading in the old one, we can buy another processor of like size. Programs that have outgrown the original processor, can then be transported over to another processor. These transported jobs continue to talk to the other jobs through Ethernet or other high speed com. The key is that while jobs have been moved from a single processor to a multiple processors, the same code is being used. Changing the node location of a program does not require rewriting the application.

There are three levels of multiprocessing support with VAXELN. The first level has been used in the previous examples and we have defined it as being distributed multiprocessing. All the VAXELN nodes are distributed across a local area network of Ethernet. Each node contains a private copy of the VAXELN kernel. In this configuration with the VAXELN runtime services and the application code this type of multiprocessing is extremely flexible, easily modified and can be fault tolerant. An example, would be three VAX computers running VAXELN, connected by Ethernet.

Jobs A and C are running on processor 1, jobs E, F and G are running on processor 2 and Jobs B and D are on the processor 3. All jobs have the ability to synchronize and send messages to each other.

Another type of multiprocessing is called tightly coupled, sometimes called symmetric multiprocessing. An example is on our 8800 where we have two processors with a single memory running a single copy of ELN. Jobs can run on either VAX processor, with equal access to kernel functions and I/O.

Another example is somewhat looser than tightly coupled but, is more tightly bound than the distributed multiprocessing; we call this 'closely coupled'. This configuration builds on a standard VAX BI computer, with KA800 modules. The KA800 is a single VAXBI module with an rtMicroVAX II processor. The KA800s can be used to control high-speed I/O modules or perform additional computational activities. The KA800 processors can communicate with each other and with the VAX primary processor over the VAXBI bus at DMA speeds for large data transfers. Using the DRB-32, data can be brought from a User Device into VAXBI memory at up to six Mbytes per second. Transfers from a User Device through the DRB-32 to KA800 private memory can occur at two Mbytes per second. The primary processor can be running either VAXELN or VMS.

VAXELN KERNEL

The VAXELN should be thought of as a set of commonly-used routines that control the VAX computer, and provide services to the application. This is contrast to the normal mode of thinking that a application runs 'under' the operating system. A VAXELN application 'owns' the VAX computer and controls the hardware through the kernel. The VAXELN kernel, also, has routines for the 'sharing' of systems resources, and provides a mechanism for synchronizing communications between jobs.

The Kernel is object oriented. That-is, it provides for process or Job synchronization by using entities known as objects. An object may be a flag, an event occurrence, receipt of a message, etc.. VAXELN processes use objects in synchronizing their execution with the events of the real-time application. Examples of objects are:

- o AREA--An area represents an amount of physical memory globally accessible by all jobs within the same system.

- o EVENT--An event represents the state of an event used for process synchronization to shared data. A typical event might be the completion of a code segment or the completion of a disk access.

- o MESSAGE--A message describes data transmitted between processes or Jobs. A message may be sent between two processes or Jobs residing on the same node or between two nodes on the same local area network or BI-Computer.

- o PORT--A port represents a repository for messages waiting to be received. Only the processes in the job that created a port can receive a message from that port;

- o NAME--Ports may be assigned a name (e.g.FRED) which is known either locally (only to jobs physically executing on a specific system, or node) or universally (to all jobs executing on an node in the local area network).

JOB AND PROCESS SCHEDULING

VAXELN Jobs and Processes are all assigned priorities by the programmer. A high priority indicates that the job or process should be given preference over other jobs and processes of lower priority when it is ready to execute. Jobs, currently executing or not, are rescheduled when one or more of its processes enter the ready state and the job's priority is higher than the priority of the currently executing job. Within that job, the process with the highest priority is given control. Job rescheduling is always pre-emptive with no round-robin scheduling or time-slicing.

REALTIME PROGRAMMING

The ability to respond to real-world events without significant delay is what makes an application a 'real-time' application. VAXELN provides a number of mechanisms to enable processes to respond to external and internal events with predictable performance.

An 'exception' is an unlikely event that occurs during the execution of a program, but is one that the programmer did not expect. Examples include division by 0, integer overflow, addressing nonexistent memory, or a power failure. Exception-handling functions can be created and established for VAXELN processes or modules. When an exception occurs, the kernel invokes the exception-handler function. This function can examine the supplied arguments to determine the cause of the exception. If the function can handle the exception, it does so, and returns a value of 'TRUE'. By returning 'TRUE', the function informs the kernel that the exception has been handled and that the process can continue execution. If the function can not handle the exception, the function returns a value of 'FALSE'. In this case, the kernel will search for a higher level exception-handling routine. If none is found, or if none can handle the exception, the process that raised the exception is deleted.

An application achieves real-time performance when it is able to respond to device interrupts within an acceptable period of time. Since VAXELN was designed with real-time performance as a principal goal, the kernel handles device interrupts quickly and predictably. Another goal of VAXELN is to make the process of building device drivers simple and easy. All device drivers for VAXELN can be built using High-Level Languages, such as VAXELN Pascal, VAX C, or VAX ADA. In most other realtime environments, device drivers must be coded largely or entirely in assembly language.

VAXELN supplies sources for the device drivers for many real-time, mass-storage, and interactive devices on the toolkit. These sources can be used for instructional purposes to build drivers for devices not normally supplied by DIGITAL. While many VAXELN device drivers run as separate jobs and communicate with calling programs, through runtime library routines and messages, normal realtime device drivers are coded as subroutines that act directly on the I/O device registers and field the interrupts. Since most realtime I/O devices do not need to be shared between jobs, this allows fast and predictable I/O without the extra overhead of unnecessary context switching.

SUMMARY

VAXELN is a unique state of the art product for the design of dedicated realtime VAX applications. With the ability for development in High-Level Languages, inclusion of Local Area Network support, and support for performance tools, VAXELN allows a more timely development cycle and helps reduce development cost.

USING ETHERNET AND FIBEROPTICS (FDDI) NETWORKS IN THE REALTIME SIMULATION ENVIRONMENT

RON RAMBIN

FS TECH CONF - ATLANTA; SEPT 1988

VISTA CONTROLS CORPORATION
27825 W FREMONT COURT
VALENCIA, CA 91355
(805) 257-4430

ABSTRACT:

This paper will present Vista Controls findings that networking via Ethernet™ is a viable and inexpensive option that can be taken advantage of by the realtime community. We discuss the example of a "Force-on-Force" realtime simulation to find the required data rates, and demonstrate that, by bypassing upper layer protocols (TCP/IP), the Ethernet LAN can meet these throughput requirements.

A discussion is then presented showing FDDI to be the most promising fiber optical interface protocol. The FDDI standard will likely become an industry and military standard very soon. Using this ring topology network for realtime applications will result in a very cost-effective and easily reconfigured multi-user networked realtime computing facility.

One concept in this area that has more and more interest, is the scenario of many networked simulators. Actual multi-engagement exercises of multiple "teams" and varying components can be simulated very realistically. For example, two fighter wings can engage in dogfights, or a helicopter squadron can take on a tank platoon. The lessons learned from this kind of exercise are invaluable, and can only come from either realtime, networked simulation, or from actual warfare. Networked simulation is a cheap, fast, painless, and accurate alternative!

1. INTRODUCTION:

In recent years, we have seen the introduction of many impressive developments in computer network performance. With these new performance enhancements, the realtime user is now finally able to seriously consider taking advantage of the many benefits of decentralized computing. Up until now it has generally only been the non-realtime user that could afford the high data and protocol overhead costs associated with any network scheme.

An effective multiple processor networking capability is always attractive for many reasons. By networking stations together, and then decentralizing the computational task(s), several benefits are realized:

1) Many small processors can be used to achieve computational tasks that would ordinarily require larger and much more powerful computers.

2) Many stations can participate in any task; both in the form of processing, as well as data I/O.

3) A single central computer is tasked with supervisory operations, and contains the total program database. The database may then be commonly shared among all network nodes. Program control, and database editing and access is then simplified. Memory requirements are reduced significantly.

4) On-line manipulation of the database and of the various programs are allowed, as well as on-line data entry at any or all of the many stations.

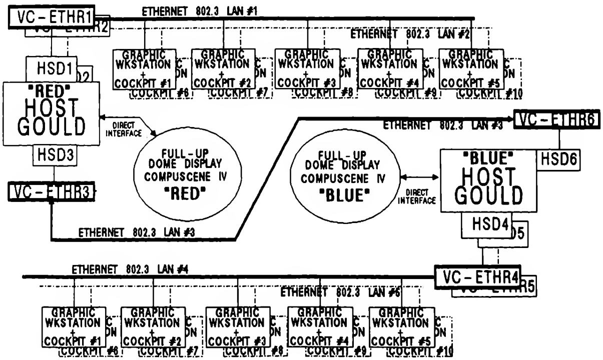
The non-realtime application can take advantage of these benefits because that user does not have to meet critical frametime constraints. On the other hand, the realtime user must always and unfailingly complete all data I/O within a frametime that is often very short. Frametimes of 10ms or even much shorter are common in most realtime applications.

This paper proposes an attractive networking solution to meet the challenges of the typical realtime user. Here, we outline an example of a "force on force" realtime simulation system (RTS) that uses a typical graphics workstation as a low cost simulation cockpit. Looking specifically at this realtime simulation task as a good example of realtime computing, we note that as the frametimes are decreased to yield greater system fidelity, overall data rates increase. In this environment, normal networking scheme capabilities are taxed beyond their limit. The data rates become significant as the frametimes decrease, and network overhead becomes a rate limiter.

We will examine two methods of data communication over networks. One method is the Ethernet LAN, and the other option is to use fiberoptic media, with the Fiber Distributed Data Interface (FDDI) protocol. The Ethernet LAN offers an inexpensive and readily available network with 10 Megabit/Second potential data throughput capability. FDDI has the advantages of being a timed, token-passing, dual-ring topology network with a very high bandwidth. In addition, fiber optics communication can easily be made a totally secure data transmission media for sensitive and classified projects.

2. ETHERNET IN RTS NETWORKS:

The realtime simulation application we investigate in this paper is the aircraft force-on-force scenario with two opposing, 11 station teams. Each force team (Red vs Blue) consists of a single "CompuScene IV-type" of dome full-up cockpit display station, and 10 additional "wing men" low fidelity stations. The auxiliary wing men are considered to be sitting at a low cost workstation, broom stick and chair type simulator. They have all basic flight and fire control systems simulated, but only with limited fidelity, and only a forward field-of-view. By studying this example, the realtime simulation user is provided with the necessary tools to estimate the true data throughput requirement for an 11 on 11, force on force realtime simulation experiment.



TYPICAL FORCE ON FORCE APPLICATION

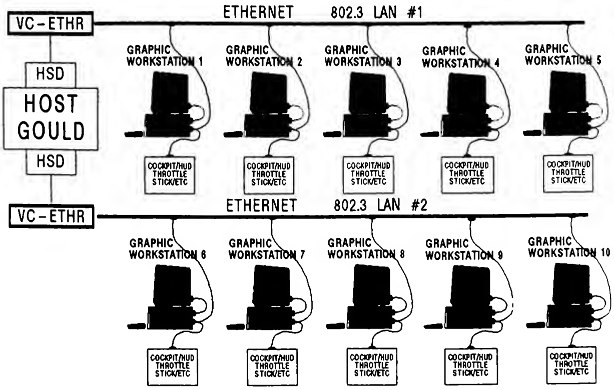
FIGURE 1

Figure 1 shows a potential set up for the typical 11 by 11, Red vs Blue, force on force set up. Each force consists of a central realtime simulation computer, a dome, and two Ethernet LANs. A "Gould" Concept/32 computer is shown as it is the most typical mainframe used for this type of program. Each dome has the primary aircraft cockpits and their associated hardware and electronics. Each Gould has two Vista Ethernet LANs hooked up, with five low cost workstation cockpits attached as nodes to each LAN. This gives a total of ten low cost workstation cockpits per Gould.

An additional Vista Ethernet link between the two Goulds is optional. This is shown on Figure 1 as Ethernet LAN #3. If used, it would provide correlation data between the two simulation scenarios. Other means of providing an effective communication link between the Goulds could be to use Shared Memory, Reflective Memory, SelNET, or other proprietary network media. However, selection of Ethernet as the intra-Gould communication network offers several unique advantages. They include allowing a wide physical separation (100 to 500 yards, or even more) between the two mainframes, and a very convenient, inexpensive way of hooking in DecNET or other networks or mainframes.

In our particular example it is assumed that each one of the Goulds contains the complete playing field. Each of the workstations has a simplified form of the playing field in a database. In this kind of a setup, each node on the Ethernet will require no more than around 60 variables each frametime. This will provide the workstation with all the data required. The entire playing field is computed and displayed, along with all the opposing aircraft models in proper perspective to the individual player. This variable count is approximate, but is based on 6DOF model of the particular node, and a 3DOF model, relative to the node, for all other aircraft. Communication of data to the full-up dome display will not be addressed in this paper, although a similar number of variables must be shipped over to it, as to the Ethernet nodes.

Each of the workstations has a simplified -own aircraft- aerodynamic representation in the force on force example. In addition, there is a minimally functional "cockpit" attached to each workstation to allow for realtime simulated pilot inputs. The cockpits are considered to have a simplified sidearm controller or motion stick, throttle, and the necessary switch panel(s) to do the most rudimentary aircraft flight and fire control functions. Fire control functions such as arming and firing guns, and firing missiles or dropping stores are also implemented. Figure 2 shows a block diagram of this kind of a system. Only the Ethernet LAN of a single Gould mainframe is shown, and not the communication links with the Compuscene or other computers.



FORCE ON FORCE ETHERNET LAN

FIGURE 2

The workstations have a simplified playing field data base in which the individual can see all of the adversaries and/or wing men presented as simplified aircraft models on the CRT. The variables being shipped between the Gould and the workstation consist of the other aircraft positions and his own position and orientation in space. The station is responsible for calculating perspective view as viewed from his aircraft. An alternative to this particular configuration is to put simplified aircraft in each of the workstations.

In this example the data rate requirements can be calculated on a per workstation basis via the following equation. (Each Gould is assumed to be running with a 10 millisecond frame rate.)

$$D_{TP} = \frac{(C_T \text{ or } T_T) + P_T + (D \times R)}{TD}$$

where:

D_{TP} = Data Throughput
T_T = Transmit Time
TD = Total Data Called or Shipped
D = Data Bytes/Call or Transmission

R = Data Rate
C_T = Call Time
P_T = Protocol Time

In this example, it is to be emphasized that we have broken down the data rate requirements to the basic node requirements. This has been done to show that there is really only a small number of data bytes to be transferred to each workstations per frame. Regardless of this low actual data throughput requirement, if a standard, highly protocoled Ethernet LAN is used, a large amount of extraneous overhead time is incurred. This overhead is caused by the mandatory handshaking and file transfer data required by Ethernet upper layer protocols such as TCP/IP. Due to the large number of data turnarounds, and by shipping small packets of data to each node per frame, the processing time as a function of packet size becomes excessive. Because of this increase in overhead time per transaction, the development of a high speed Ethernet "Data Highway" ™ is necessary for this realtime application.

In this force on force example we have made several assumptions in calculating the throughput requirements - detailed as follows:

1) The workstation presentation is primarily a CRT representation of looking through the Head Up Display (HUD) at a very limited field of view. The pilot may be able to see all other planes (adversaries and friends), and the flight/fire control data that would normally be presented on the HUD.

2) Each workstation has a simplified playing field database imbedded in it. All that is necessary is that the vectors of ownship and other vehicles positions are given to the terminal. The workstation does all the kinematical calculations and/or positional calculations for display.

3) The data coming from or going to the workstations must occur in the first few milliseconds or the last few milliseconds of each frame. This is particularly important if current information is to be used for the calculations so that there is no frame skew in the calculation nor in the presentation. Sampling time to the main processing cycle depends on the terminal CRT mode of whether it is a 60 HZ non interlace update display or a 30 HZ interlace upframe display.

4) All ownship calculations are performed locally, at the Ethernet node. This includes all acceleration and position vectors, and further implies that the only data that must be shipped back to the Gould host are the three position, velocity, and acceleration vectors.

3. ETHERNET THROUGHPUT:

We will now examine some of the data rate problems that occur when trying to use the Ethernet LAN to couple together these low cost workstations. As stated earlier, we generally have to ship around 60 variables (actually - 63 in, 3 out) per frame to provide a single station with the necessary vector information for an 11 on 11 force engagement. Our company has had this experience in several previous related simulation jobs. Fifteen bit digital accuracy plus sign bit yields a total of two bytes per variable. This comes out to a total of roughly 120 bytes per station, per frame, shipped to the node.

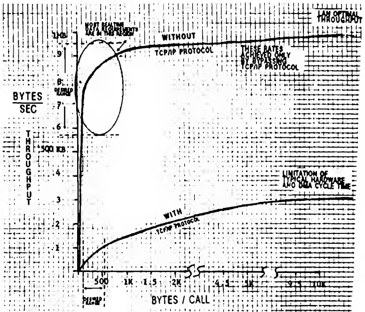
In a current simulation task we are working on, a 10 millisecond frametime is used. These factors would imply a "gross" total data throughput of only 12 Kbytes per node, per second. However, this figure ignores the necessity that the workstation must obviously have enough time during the frame to input, perform calculations, and present the new data. Therefore, the station has an "I/O window" that is determined by its computational capability and required tasks. In a typical force on force application, as described in this paper, all I/O to and from the workstation will occur during either the first or last millisecond of the 10 millisecond frame.

By taking into account this I/O window requirement, the actual effective required data rate comes out to be 120 Kbytes per node, per second. With five nodes on the LAN, then roughly 600 Kbytes per second total data throughput is the minimal requirement (per LAN) for a twenty station force on force engagement. This number can be bumped up to as high as 1 Megabyte, and still be supported by a raw Ethernet "Data Highway".™ That leaves plenty of room for additional variables that may be unique to the application, and it also leaves some overhead room to allow for the processing speed of the stations in receiving data. Very often, at these data rates, we find that the stations themselves are rather slow, and special software I/O drivers have been written to solve these problems.

In a particular example of this kind now being worked by Vista Controls, each workstation is a Silicon Graphics "IRIS" workstation. The IRIS is a VME-based system with a standard plug-in Ethernet interface card by CMC. The CMC card supports the 802.3 level of Ethernet fully, and with the addition of some low level drivers, these data rates are achieved. The software that must be written is to expedite the data calls in and out of the CMC Ethernet Controller. These routines are standard, and generally do not change with different applications.

For a more complete picture of the typical data throughput requirements, and how the upper layer protocol layers affect those rates, we are presenting Figure 3. This figure shows the actual throughput rates that can be achieved with an Ethernet interface, both with and without those upper layers. Note the shaded performance area near the top left of the graph. That area defines the type of performance that is generally required by a realtime simulation user. The number of bytes per call to be sent to each node will vary from around 50 to 500. The number of stations on the network will vary from 2 to 20. Frametimes will normally vary from 50ms down to 1ms.

Two traces are shown on the graph. The lower trace was obtained by measuring a typical Ethernet interface card, which uses the full TCP/IP protocol. The top trace was obtained from doing the same performance analysis on the raw Ethernet LAN connection. It can be clearly seen from this graphical representation that an Ethernet LAN, which implements full protocol, will not be very satisfactory. Either it will be a LAN in name only, and will be limited to only one or, at the most two nodes, or the amount of data transferred per frame, per node will be severely limited, or the frametimes must be increased to more on the order of 500 to 100ms.



ETHERNET DATA THROUGHPUT

FIGURE 3

None of these options are desirable for the realtime user. The last option will often take the whole problem right out of the realtime world. On the other hand, if the amount of data transmissions is severely constrained, then system fidelity suffers. And finally, if the "dedicated" LAN "solution" is chosen, it usually turns out to be a very costly solution in both engineering time and money.

Note that the upper trace offers a much more attractive solution. When the Ethernet LAN can be used at, or near its rated capabilities, the realtime user is able to achieve performance levels that are exactly in concert with current requirements. We can see that there is more than enough LAN bandwidth to fully support as many as five to eight stations successfully in this kind of a force on force scenario. For other types of realtime applications, it is not unreasonable to consider upwards of 15 to 25 networked stations, consisting of all kinds of node devices that have Ethernet connectivity. The network may include multiple PC's or AT's, graphic workstations, or any other type of device with an Ethernet interface. The reader is asked to keep in mind that these rates are measured as MEMORY TO MEMORY rates, from/to Gould host, to/from workstation. As discussed, they can be achieved only by bypassing all the upper layers of protocol, and sending out raw Ethernet over the wire.

Vista Controls has developed an interface to achieve this kind of high speed Ethernet LAN communication for the Gould computer. Other applications may require development of a similar interface to different host machines. In any case, the approach is the same. Most computers have a High Speed Data, or a direct CPU interface of some kind which supports data transfer by Direct Memory Access (DMA). It is usually a 32 bit parallel I/O interface. On the Gould computer, the "HSD" supports DMA data transfer at rates up to one MBPS in the external mode, with virtually no host CPU overhead at all. Therefore, a direct HSD/Ethernet interface was developed to support the shown data rates.

At the other end of the LAN are all the nodes. We have found that virtually all of the common devices that would be considered for this kind of an application offer Ethernet connectivity as an off-the-shelf support accessory. Most of the potential nodes, such as Sun workstations, IRIS's, Apollo's, etc. use a standard Ethernet interface to their own system such as the ones offered by Exelan or CMC. In some cases, a proprietary interface from Ethernet to the node is used. But in virtually every instance we have investigated so far, the hooks are already there in the on-board software or firmware to support the low level calls that are needed by the raw Ethernet transmission.

That being the case, the realtime user now has readily available a high speed Ethernet LAN potential that implements only the low level protocol on both ends. Through software optimization of the drivers, faster and higher data rates at both ends of the problem can be developed.

4. FIBER OPTICS (FDDI) IN RTS:

Fiber optical media offers the realtime user a set of advantages that cannot be matched by any other networking scheme. For one thing, the bandwidth of FO is a factor of 5 to 100 times as wide as most other hard-wired electrical bussing techniques. In addition, FO offers a totally secure data transmission media.

There are presently several types of FO specifications now being promoted. However, we at Vista Controls are convinced that the Fiber Distributed Data Interface (FDDI) will become the most widely selected FO bus for future military and avionic applications. It has already been tagged as a NASA defacto standard, is now being used by Air Force ATF contractors as a likely "fly by light" candidate, and has been selected for future MODSIM program work.

The FDDI protocol has a 100MBPS bandwidth at peak. It would appear that FDDI is therefore 10 times faster than the Ethernet systems described above. Even though that is true from a strict data rate point of view, it turns out that in the Gould computer environment, only an improvement on the order of a factor of five is possible. This limitation appears to be only a current, temporary barrier to full exploitation of FDDI bandwidth that Gould, as well as other mainframe computer manufacturers are addressing. The only reason for this constraint in I/O performance is the minimum DMA data transfer cycle times that are now available on the existing Gould HSD interface card.

Even with this temporary improvement limitation, as many as fifteen to twenty-five terminals or nodes could be operational per FDDI LAN rather than the roughly five node limit on the Ethernet LAN. At these rates, this provides a way to utilize the Gould maximum I/O power in communicating with external devices. This also provides an extremely fast data transfer between Goulds and other mainframe type computers such as the DEC, Harris, and others.

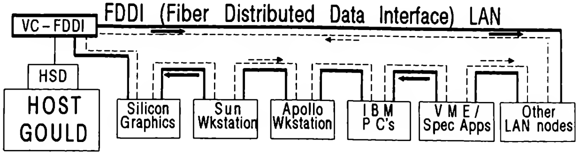
By designing present simulation facilities around the developing FDDI fiber optical LAN protocol, the potential bandwidth exists that will be necessary to drive graphic systems at rates that are presently unattainable. Data rates in the five to ten megabytes per second can be easily achieved using current technology. The newer graphic workstations will be able to handle this data stream. They will thereby concurrently achieve the desired goals of higher system fidelity, faster framing rates, and /or more terminals per network.

Figure 4 shows a block diagram of how a typical FDDI simulation facility could be configured. A dual ring topology is shown (redundant bussing), but is not necessary. One major difference between the Ethernet and FDDI is that FDDI does require at least a single ring topology, rather than being an open-ended network structure. It operates via a timed, token-passing protocol.

FDDI has the support of most major mainframe manufactures and the largest chip manufacturers. It seems clear that within a short time, we can expect to see a large collection of available FDDI products. Many of these will be designed to support the realtime simulation user, and a good percentage of them will become available over the next year.

Either of these networks offer a very cost effective manner to achieve the multi-engagement capability. The proliferation of available low cost graphics workstations allows good fidelity of presentation and provides a quick and efficient way to integrate an additional cockpit into the force on force system. The use of low protocol networks allows for ease of integration while maintaining the high data rates.

We expect to see many networked applications of realtime flight simulation programs in the future. Most of them will likely evolve from either or both Ethernet and the various fiberoptics interconnect networks. As the data rates go from one megabyte per second to ten megabytes per second, many peripherals can be tied on to the mainframe simulation to add additional fidelity and additional elements of simulation.



TYPICAL FDDI
LAN USAGE

FIGURE 4

5. CONCLUSIONS:

The purpose of this discussion has been to demonstrate the following:

1) An Ethernet "Data Highway"™ LAN, implemented without the higher level protocols, offers the realtime simulation user a low cost, high throughput network medium. The typical "force-on-force" simulation is an excellent example of how to implement a cost-effective, optimized realtime networked facility.

2) FDDI is the recommended fiber optic media for the avionic world of the future. It will yield very high data rates, while maintaining an unsurpassed level of data security and integrity.

The airframe designer will be able to evaluate in a true force on force environment the particular advantage or disadvantages of any new or developmental hardware or software in his airframe. As an example, one could actually test in a realtime, simulated warfare environment whether or not the addition of new radar with new capabilities would add significantly to the ability to entertain and defeat a certain foe. The importance of having the capability to evaluate new concepts in a true combat environment cannot be overemphasized for future programs. Networked simulation facilities offer this important capability at low cost and are easily integrated.

We conclude that the use of networks in the flight simulation environment is long overdue. The application of these very high speed data networks will allow a much more flexible, and much more easily reconfigured and lower cost force on force implementation in the future.

™ Data Highway is a trademark of Vista Controls Corp.
™ Ethernet is a trademark of Xerox Corp.
™ SelBUS, MPX-32 and UTX/32 are trademarks of Gould Inc.
® CONCEPT/32 is a registered trademark of Gould Inc.

SOME BENEFITS OF DISTRIBUTED COMPUTING ARCHITECTURES FOR TRAINING SIMULATORS

William B. Forbes
Principal Software Engineer
and
Jeffrey B. Kauffman
Senior Software Engineer

Digital Equipment Corporation
Laboratory Data Products Group
Marlboro, MA 01752

Abstract

Given the asynchronicity of external I/O in a simulator environment, a simulator designer is faced with a task of creatively utilizing system resources to effectively handle the external I/O. To ensure a realtime response, sufficient margin of performance has to be provided by classical computer system architectures. The margin has to be high enough to allow for the worst case design: the simultaneous occurrence of asynchronous events. This results in an underutilized resource, or, effectively an "overbought" system. Tightly coupled multi–computing architectures offer the benefit of allowing the designer to dedicate computing elements to specific tasks which occur asynchronously. By matching the modularity of the system design to the modularity of the application, the problems associated with resource management can be greatly simplified. The same architecture allows for more cost–effective system adaptations to re–specified external I/O requirements. This paper describes several variations of a technical solution which designers can take advantage of when using multi–computing architectures to increase the determinacy of simulator response. The primary focus will be on designs which use front–end processors to offload asynchronous I/O setup and completion tasks from the main system processor(s). In addition, the utility of front–end processors in reducing the data traffic in and out of system–wide shared memory and the attendant increase in bus bandwidth availability will be discussed.

Introduction

The simulation of flight and of aircraft systems is a fundamentally modular problem. The system being simulated is itself modular, consisting of multiple line replaceable units (LRUs) and other discrete components – airframe, weather, radar, threat, etc. Furthermore, each component of the simulation has distinct input, computational, and output phases. For these reasons, computer systems for flight simulation are often modular, so that the computer design can match the modularity of the application.

This paper presents a discussion of modular designs for flight (avionics) simulation systems which incorporate heterogeneous, closely coupled general purpose processors. The architecture to be discussed is heterogeneous in that the processors fall into two distinct classes which are differentiated both by their hardware and software attributes. It is closely coupled in that all processors share a common physical address space, including both memory and I/O areas (with certain restrictions, which are discussed below). The processors are general purpose in that all are multi–tasking, all are user programmable in high–level languages, all share a common development environment, and all are capable of performing I/O functions as well as computational tasks.

The goal of this architecture is to provide a high degree of flexibility in regard to the distribution of sub–tasks among processors of each class, based upon the hardware and software attributes of the two classes of processors. One class of processors, the "system" processors, is relatively high performance, shares a common memory interconnect, and runs under a tightly coupled multiprocessing operating system (VMS ™). A second class of processors, the "front–end" processors, is relatively lower in performance, has local memory for program and data storage, and runs its own copy of a low overhead realtime executive (VAXELN). In general, the system processors are well suited for performing synchronous computational tasks, while the front–end processors are well suited for performing asynchronous I/O tasks.

Resource Planning

In specifying a computer configuration for a flight simulation system, the system designer is confronted with the following resource planning questions:

- Does the configuration have adequate hardware resources to meet the performance demands of the application? (e.g. Does the configuration have adequate I/O bandwidth (single–channel and aggregate), are the processors capable of handling the compute load of the models, etc.)
- What proportion of the resources of the system will be consumed by software overhead?
- Will system resources be available when needed to service event–driven tasks?

To address these resource planning problems, designers often choose multiprocessor (MP) architectures. Conventional MP designs very effectively address the first part of the resource planning problem – getting enough hardware resources. If it is found that two processors aren't adequate to carry the compute load, another processor or two can be added. The exercise of this option to add more processors is facilitated by the fact that the modularity of MP designs is well matched to the modularity of the application, as described above.

Conventional MP designs offer only limited help with the other two resource planning problems – control of software overhead and guaranteed availability. In theory, increasing the number of processors in a system gives the designer greater ability to dedicate processor resources to high priority sub–tasks. This improves the predictability of system responsiveness by reducing contention for processor resources. However, if general purpose processors are used, their cost often prohibits their being dedicated to the performance of a single, high priority task. Large, general purpose processors are not well matched to small, time–critical tasks. Alternatively, small, dedicated I/O processors may be used to serve specific time–critical I/O tasks. This approach can result in a good match between the cost and size of the processor and the specific tasks it is designed to perform. However, conventional I/O processors (microprocessor based, typically interfaced through a parallel interface) have limited programmability and are therefore only applicable to a narrow range of I/O sub–tasks. This constrains the designer in how I/O processor resources can be used.

Realtime Accelerator Architecture

The new architecture described here combines the advantages of both a general purpose MP approach and a dedicated I/O processor approach to resource planning. It allows configurations to be constructed which incorporate both large, high performance processors and small, low cost processors. Both classes of processors are fully programmable in several high–level languages, giving the designer great flexibility in how tasks are distributed among processors.

The example we are going to give here is based on VAXBI bus systems such as the VAX 8250, 8350, 8700, 8800 etc. The recently announced VAX realtime accelerator (RTA) processor is combined with a system cpu(s) (e.g. VAX8800) and I/O devices such as the DRB32 parallel interface. Using this architecture the RTA processor can be dedicated as a front–end processor.

The key component of this new architecture is the VAX RTA processor. The architecture of the processor is based on the MicroVAX II design in a VAXBI bus form–factor. The performance of the processor is equivalent to a MicroVAX II class machine.

The RTA processor is a single–board option with 1 MByte of local memory onboard. It has a local memory interconnect providing for up to 5 additional 2 MByte memory arrays. There is a slave port to the VAXBI bus which allows other processors or devices on the same bus to address the local memory of an RTA processor and allows the RTA processor to access the full address space of the VAXBI bus. The interface to the VAXBI bus also supports the full range of transaction types defined for the bus. In particular, the RTA processor is capable of receiving interrupts from I/O devices or other options on the bus and fully supports the "multi–responder interrupt" capability of the VAXBI bus.

In system configurations which use a single VAXBI bus as both an I/O bus and a memory interconnect between the system processor(s) and the system memory array (for example, the VAX 8350), the full address range of the system is accessible by all processors, including any RTA processors. In these configurations, the memory address range for the RTA processors' local memory is contiguous with the address range for the system memory array. (Note: The 1 GbyteVAX physical address space is equally divided between memory space, in lower part of the range, and I/O space, in the higher part. Normally, all memory is configured contiguously in the lower half of address space, though in certain configurations, memory arrays can also be configured in reserved portions of I/O space.) Some system configurations make use of a high–speed memory interconnect between the system processors and the system memory array (for example, the VAX 8800). In these configurations, VAXBI bus adapters are used to attach multiple VAXBI busses which serve as I/O busses. The VAXBI bus adapter propagates address references in I/O space from the memory interconnect to the I/O bus and it propagates address references in memory space from the I/O bus to the memory interconnect. This implies that data areas to be shared among system processors and RTA processors can be configured either in the system memory array or in a memory array in I/O space (see Figure 1).

The system processors execute a tightly coupled multiprocessing operating system (VMS). System data structures and operating environment are held in common among all system processors. The RTA processors execute a multi–tasking realtime kernel (VAXELN). Each RTA processor executes its own set of task images. The initialization, loading, and running of VAXELN images in the RTA processors is controlled by processes executing under VMS in the system processors. In addition, a library of callable routines for communication and synchronization among system and RTA processors is provided (see Table 1). The programmer has the option of using these routines for interprocessor communication or of simply sharing data among processors, as described above.

The runtime environment of an RTA processor under the VAXELN kernel is entirely memory–resident. All code and data are contained in physical memory unless explicitly written out to disk by a user process. The VAXELN runtime kernel occupies 28 KBytes of local memory, leaving 973 KBytes of physical memory for user programs and data on an RTA processor with no extra local memory options. Multi–task system images targeted for RTA processors are compiled, linked, and built under VMS and then loaded into the RTA processor(s). All the program development tools supported under the VMS operating system, including language sensitive editors and code management utilities, can be used to develop code targeted at an RTA processor. Language support is provided for Ada® , C, FORTRAN, Pascal, and VAX MACRO–32 assembly language. Because of the memory–resident nature of VAXELN images, device

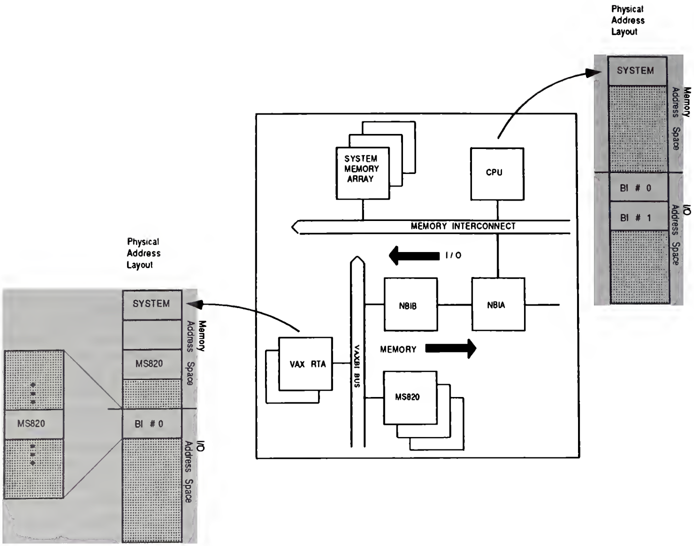


Figure 1: VAX RTA – VAX 8800 Configuration

drivers may be coded in any high–level language except FORTRAN.

In the following sections, an example application will be described and an implementation of the new architecture will be shown, illustrating its benefits.

Model application

The example application to be considered is an avionics simulation system, comprising both remote terminal simulation and environmental simulation with remote terminal stimulation. A block diagram of the simulator is shown in Figure 2.

In this example, a single 1553 bus is used. Two remote terminals (RTs) are simulated and two are stimulated with either analog or discrete digital signals derived from the airframe simulation. The two RTs which are stimulated output analog and discrete signals which must be read into the simulation computer each time their inputs are updated. The major and minor cycles of the avionics system are 100 msec (10 Hz) and 33 msec (30 Hz), respectively. The frame time of the airframe simulation is 16.7 msec (60 Hz), and is asycrhonous with respect to the avionics cycles. The simulation computer supports an operator console which monitors the simulation graphically and alphanumerically with a 1 sec update period. The operator can inject error conditions for either the simulated RTs or the sensors of the stimulated RTs by entering error codes at the console. A bus monitor is used to extract traffic to and from the simulated and stimulated RTs for later analysis. The data volumes and rates are shown in Table 2.

Simulation Computer Configuration

The simulation computer configuration used to illustrate the use of the RTA processors in the example application is shown in Figure 3.

In this example configuration, analog and discrete digital I/O to and from the stimulated RTs is performed by an analog and digital stimulation sub–system through a DRB32–W parallel interface. Traffic between the 1553 bus and the simulation computer is handled by a controller comprising a bus interface unit (BIU) and a dual–ported memory (DPM). The DPM is configured in I/O address space and is therefore visible to all processors in the system and to the BIU. The simulation system places data in the DPM which the BIU incorporates into messages in response to status requests to various sub–addresses of the two simulated RTs. All traffic to and from the operator console is handled by a 9600 baud asynchronous serial controller.

Table 1 Control and Communication Routines

Routine	Function
Control Routines	
VRTA$GET_SYSTEM	Obtains system information about an RTA processor
VRTA$INIT	Initializes an RTA processor
VRTA$LOAD_PROGRAM	Loads an executable task image into a running RTA processor
VRTA$LOAD_SYSTEM	Loads an executable system image into an RTA processor
VRTA$SHUT	Shuts down an RTA processor
VRTA$START_JOB	Runs a task in an RTA processor
VRTA$UNLOAD_PROGRAM	Removes an executable task image from an RTA processor
Communication Routines	
VRTA$ACCEPT_CIRCUIT	Establishes a circuit between two ports
VRTA$CONNECT_CIRCUIT	Connects a port to a specified destination port
VRTA$CREATE_MESSAGE	Creates a message and its associated message data
VRTA$CREATE_NAME	Creates a name for a port
VRTA$CREATE_PORT	Creates a message port
VRTA$DELETE	Removes a message, name, or port object
VRTA$DISCONNECT_ CIRCUIT	Breaks the circuit connection between two ports
VRTA$JOB_PORT	Returns the current job port
VRTA$RECEIVE	Receives a message from a port
VRTA$SEND	Sends a message to a port
VRTA$TEST_ALL	Tests for messages at all of the specified ports
VRTA$TEST_ANY	Tests for messages at any of the specified ports
VRTA$TRANSLATE_NAME	Translates a character string into a port value
VRTA$WAIT_ALL	Waits for messages at all of the specified ports
VRTA$WAIT_ANY	Waits for messages at any of the specified ports

Table 2 Data volumes/rates for example application

Data Source	Data Throughput Sim = >Avionics	Data Throughput Avionics = >Sim	Update Period	Interrupts per Update
Stimulated RTs (1553 BIU)	20 bytes	100 bytes	50 msec	2
Simulated RTs (DRB32-W)	800 words	200 words	33 msec	1
Console output	400 bytes	1-2 bytes	1 sec	1

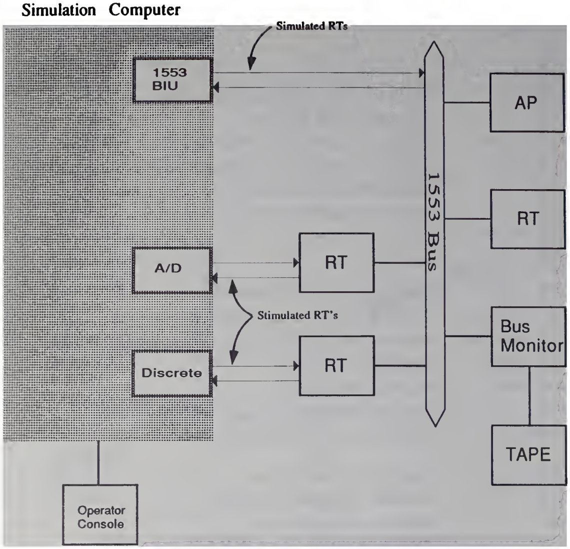


Figure 2: Simulator Block Diagram

The system CPUs execute the airframe simulation computations. A current value table for the simulation is maintained in the system memory array. In addition to containing the results of intermediate calculations for the simulation, this table is the source and sink for data flowing between the simulation and the operator console, the stimulated RTs and the 1553 environment. The 16.7 msec frame cycle of the simulation computer is driven by a realtime clock. This clock is not synchronized with the major or minor cycles of the avionics system.

Without RTA processors, the designer would face several resource planning issues:

1. Colliding Interrupts. In addition to the clock, three devices are able to generate asynchronous interrupts. What if all 4 devices interrupted at the same time? There must be adequate processor resource in the system to service all 4 interrupts within some critical time–window while at the same time keeping up with the normal computational requirements of the 16.7 msec simulation frame period.
2. I/O Processing Overhead. The overhead associated with managing the flow of data through the drivers for the serial and DRB32–W devices and, to a lesser degree, the task of updating the status data in the DPM must be determined and taken into account. Some frames of the simulation are "busier" than others in this regard, since all I/O channels operate on slower cycles (Synchronized with the simulation cycle) than the simulation, in this example. However, the design must accommodate the worst case (i.e., the busiest cycle), with a resulting excess of processor cycles during cycles in which no I/O is required.

In the example configuration, two RTA processors are used to manage the flow of information between the operator console, the simulation computer, the stimulation sub–system, and the avionics bus. One of the RTA processors is used to control the stimulation sub–system and serial channels. Both of these devices would be capable of performing DMA transfers to or from either the local memory of the RTA processor or the system memory array.

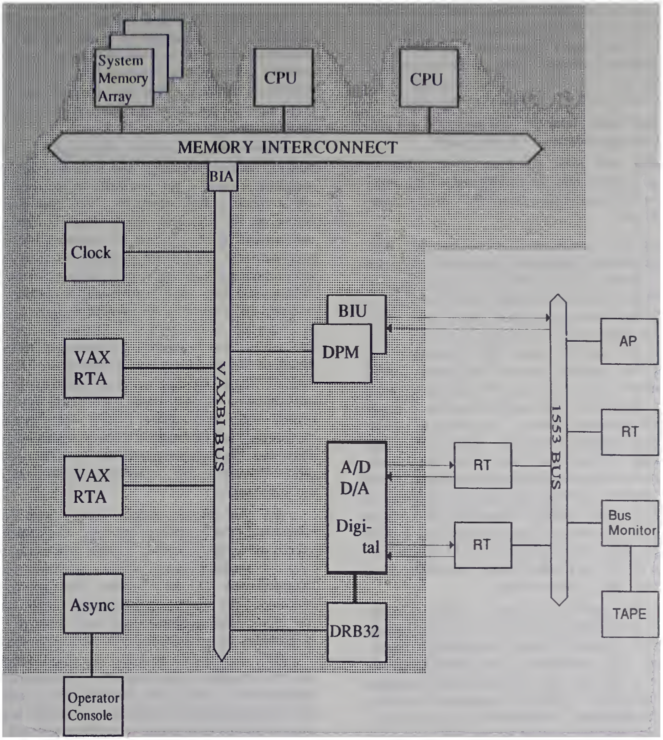


Figure 3: Simulation computer configuration using RTA processors

In the case of the operator console, the RTA processor could execute a set of tasks which performed the following functions:

- Format a subset of the current value table and write it to the console screen in either graphic or alphanumeric form at a 1–second update rate
- Capture keyboard input and modify the format of the console screen accordingly (e.g. select different variables to monitor)
- Capture keyboard commands and inject simulated error conditions by modifying the simulation current value table
- Capture keyboard commands and manipulate simulation parameters (e.g. start/stop simulation models)

In the case of stimulation sub–system , the same RTA processor could perform the following functions:

- Copy stimulation data from the current value table into local memory; perform any necessary engineering unit or format conversions on the data and package it into appropriate messages to the Stimulation sub–system (DRB32–W).
- Write command messages to the analog and/or digital devices requesting conversions on the command/status channels of the stimulated RTs
- Capture the analog and/or digital command/status data; perform engineering unit conversion; check for over–limit and copy to the current value table

A second RTA processor could be used to manage the flow of data between the simulation computer and the avionics environment. Since the timing requirements for this channel are more stringent than for the console and DRB32–W channels, a dedicated processor resource for this channel alone is called for. The RTA processor dedicated to the 1553 channel could perform the following functions:

- Accept an interrupt from the realtime clock and synchronize the other processors in the system
- Every other cycle, move 800 bytes of data from the simulation's current value table to the dual–ported memory; move 200 bytes of data from the DPM to the current value table

By distributing these various I/O tasks to dedicated RTA processors, the designer insures that worst–case contention for processor resources will not result in missed response deadlines. Without the RTA processors the system CPU(s), given a sufficiently heavy simulation model load, could possibly slip a frame while an external event was being serviced. In this example, each RTA processor fields interrupts from two different devices. This means that the worst case response latency for each RTA will be extended by an interval equal to the time required to service a single blocking or pre–empting interrupt. The ability to tolerate a single blocking interrupt in the RTA which services the clock and avionics devices is improved by the fact that the multi–tasking system kernel used by these processors allows much greater flexibility in what sub–tasks are performed at the level of the interrupt service routine (ISR). This means that the time–critical code for, say, responding to a mode command from the avionics processor could be done at ISR level and so only be blocked for the time required to execute the clock ISR in the worst case.

Although the time required to execute an ISR is highly dependent on the attributes of the drivers, it is typically not more than 100 microseconds before a driver forks to enable other interrupts to be processed. If the demands of the application were such that even that much extra response latency was intolerable, a third RTA processor could be added to the configuration. Alternatively, the most time–critical interrupt could be serviced by one RTA processor and the other three could be serviced by the second RTA processor.

However the interrupts are distributed among RTA processors, the system processors will exhibit much more predictable performance in the computation of the airframe models since they are relieved of all asynchronous event–driven processing. The activity of the RTA processors is almost completely transparent to the CPUs running the models. New data and/or flag settings appear in the system's current value table and updated parameters are drawn from that table and passed outward to the operator console and the avionics environment without any involvement on the part of the system processors.

Conclusions

In the preceding sections, it has been shown that using dedicated RTA processors to handle the I/O processing requirements of an avionics simulation application can produce substantial improvements in the predictability of system performance. First and foremost, this architecture helps meet the performance criteria of demanding realtime applications.

In addition, this architecture offers significant benefits in terms of design flexibility. The RTA is a small, low cost processor running a low overhead realtime system kernel and as such is well suited as a dedicated processor for low–level, time–critical tasks. In this respect it is like conventional IOPs. Unlike conventional IOPs, the RTA processors are true general purpose processors in that they are programmable in high–level languages, they are multi–tasking, and they conform to a well established processor architecture. These features give the designer a high degree of discretion in how the RTA processors are used in a design. Where is the boundary between the "front–end", I/O functions and the "back–end" computational functions of a system? Should the front–end processor do engineering unit conversion on the data stream it manages? Limit checking? Should it turn short control loops? With this architecture, the designer has the freedom to draw the boundary between front–end, I/O processing tasks and back–end computational tasks wherever is best, depending on the requirements of the application.

The ability to take advantage of this design flexibility is enhanced by the fact that the software development tools are consistent across processor classes. The same compilers, language sensitive editors and configuration management tools are used. The fully symbolic debugger for the RTA processors presents a user interface nearly identical to that of the host processor's symbolic debugger. Software is provided for controlling the runtime environment of the RTA processors at both command language and process execution levels from the system processors. A complete set of utilities for interprocessor communication and synchronization is available. In short, the software support for this architecture is complete and well integrated with the proven software environment of the base system.

The architecture is flexible, expandable, and user–friendly, providing an exceptional platform for distributed designs in high–performance realtime applications such as flight simulation.

™ VAX, VAXBI, VAX RTA, VMS, and VAXELN are trademarks of Digital Equipment Corporation.